

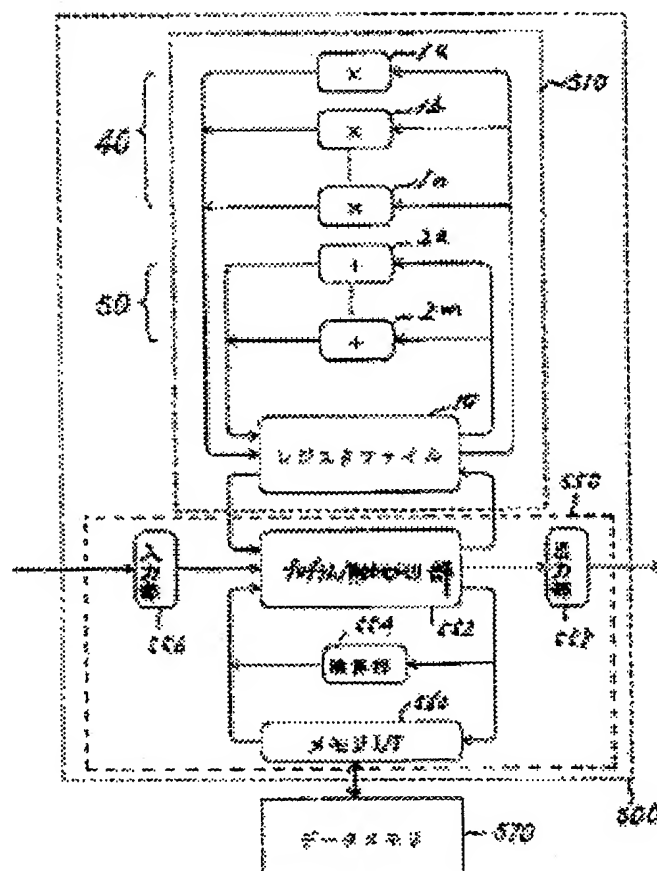
# DATA DRIVING TYPE INFORMATION PROCESSOR

Publication number: JP6176177  
 Publication date: 1994-06-24  
 Inventor: OKAMOTO TOSHIYA  
 Applicant: SHARP KK  
 Classification:  
 - International: G06F15/82; G06F15/76; (IPC1-7): G06F15/82  
 - European:  
 Application number: JP19920326742 19921207  
 Priority number(s): JP19920326742 19921207

Report a data error here

## Abstract of JP6176177

**PURPOSE:** To execute a program including an arbitrary filter processing at a high speed. **CONSTITUTION:** The device, includes a data driving type processor 550 and a data driving type filter 510 executing the filter arithmetic operation to a data driving system according to the instruction from the data driving type processor 550. The filter 510 starts its arithmetic operation by accepting an instruction in a function reading system from the processor 550 and includes a register file 10 including plural registers, 1st arithmetic unit group 40 including dividers 1a to 1n, and 2nd arithmetic unit group 50 including addition and shift composite arithmetic units 2a to 2n. The register file 10 includes a program control part which reads out instructions required for the corresponding filter processing according to the instructions given from the processor 550 and control the data and bus connection in the register file. The data of each register of the register file 10 are given to the arithmetic unit groups 40 and 50 according to the program, executing the filter processing.



(19)日本国特許庁(JP)

(12) 公開特許公報(A)

(11)特許出願公開番号

特開平6-176177

(43)公開日 平成6年(1994)6月24日

(51)Int.Cl.<sup>5</sup>

G 0 6 F 15/82

識別記号

庁内整理番号

7323-5L

F I

技術表示箇所

審査請求 未請求 請求項の数1(全 44 頁)

(21)出願番号 特願平4-326742

(22)出願日 平成4年(1992)12月7日

(71)出願人 000005049

シャープ株式会社

大阪府大阪市阿倍野区長池町22番22号

(72)発明者 岡本 俊弥

大阪府大阪市阿倍野区長池町22番22号 シ

ャープ株式会社内

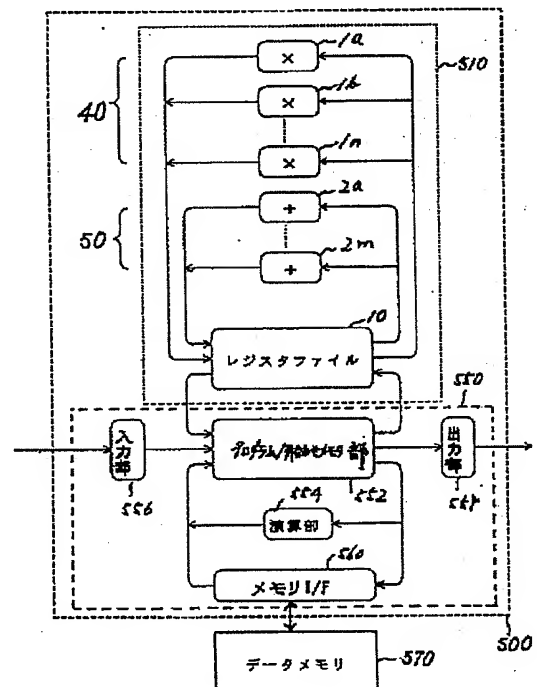
(74)代理人 弁理士 深見 久郎

(54)【発明の名称】 データ駆動型情報処理装置

(57)【要約】

【目的】 任意のフィルタ処理を含むプログラムを高速で実行することのできる情報処理装置を提供する。

【構成】 データ駆動型情報処理装置は、データ駆動型プロセッサ(550)と、このデータ駆動型プロセッサからの命令に従ってデータ駆動方式へフィルタ演算を実行するデータ駆動型フィルタ装置(510)を含む。このフィルタ装置は、プロセッサから関数読出の形式の命令でその演算が起動される。フィルタ装置は、複数のレジスタを含むレジスタファイル(10)と、除算器(1a~1n)を含む第1の演算器群(40)、加算およびシフト複合演算器(2a~2n)を含む第2の演算器群(50)を含む。レジスタファイル(10)は、プロセッサから与えられた命令に従って対応のフィルタ処理に必要な命令を読み出し、レジスタファイル内のデータおよびバス接続を制御するためのプログラムコントロール部を含む。レジスタファイルの各レジスタのデータがプログラムに従って第1および第2の演算器群へ与えられてフィルタ処理が実行される。



## 【特許請求の範囲】

【請求項1】 複数のレジスタを有するレジスタファイルを備え、前記レジスタファイルは、処理すべきおよび処理後のデータを格納するデータフィールドと、前記データフィールドにおけるレジスタを特定する行先情報を格納する行先フィールドとを含み、かつ前記レジスタファイルの1つのエントリはデータフィールドに含まれるデータレジスタと行先フィールドに含まれる行先レジスタとを有し、乗算器と加算シフト複合演算器とを含む演算手段と、処理されるべきプログラムに付されたノードに対応して命令を格納するプログラムメモリ手段と、オペランドとノード特定番号とを少なくとも含むデータパケットに応答して、前記プログラムメモリ手段へアクセスし、該ノードに対応する命令を読み出す命令読出手段と、前記命令読出手段により読出された命令が前記演算手段を用いる演算を指定しているか否かを判別する判別手段と、前記判別手段の判別結果が前記演算手段の利用を示すとき該読出された命令と、この命令とリンクして読出されたノード特定番号とオペランドデータとを含むデータパケットを送出する送出手段と、前記送出手段からのデータパケットに応答して前記レジスタファイルのデータフィールド内の所定のレジスタへオペランドデータを書込むデータ入力手段と、前記送出手段からのデータパケットに応答して、前記レジスタファイルの前記データフィールド内のレジスタ内容を読み出し前記演算手段へ与えるデータ読出手段と、前記データパケットの命令に응答して、前記演算手段の出力を前記データフィールド内のレジスタへ書込むデータ書込手段とを備え、前記データ書込手段は前記行先フィールド内の行先レジスタが記憶する行先情報が示すデータレジスタへ前記演算手段の出力を書込む手段を含み、前記送出手段からのデータパケットに응答して、前記レジスタファイルの前記データフィールド内の所定のデータレジスタの内容を読み出し、データパケットの形態で前記命令読出手段へ与えるデータ出力手段とを備える、データ駆動型情報処理装置。

## 【発明の詳細な説明】

## 【0001】

【産業上の利用分野】 この発明はデジタル信号を処理するための装置に関し、特に、映像信号および音声信号等のデジタル信号をデータ駆動方式に従ってフィルタ処理を含むプログラムを実行するデータ駆動型情報処理装置に関する。

## 【0002】

【従来の技術】 信号処理の分野においては、正確に所望の信号処理を実行するために、波形整形、ノイズ成分の

除去および所望の信号成分の抽出などが行なわれる。このような処理のためにはフィルタが用いられる。フィルタには、受動素子（インダクタンスL、キャパシタCおよび抵抗R）を用いるLCRフィルタ、演算増幅器および抵抗RならびにキャパシタCを用いる能動フィルタがある。このLCRフィルタおよび能動フィルタはアナログ信号をアナログ形態でフィルタ処理しており、アナログフィルタと呼ばれる。

【0003】 一方において、映像信号および／または音声信号などを処理する分野においては、信号処理の精密化、容易化および高度化などのために、信号はデジタル形態で処理されることが多い。この場合、アナログ信号がデジタル信号に変換された後にこのデジタル信号に対し所望の信号処理が実行される。このようなデジタル信号処理分野においても、ノイズ成分の除去および所望の成分の抽出などのためにフィルタ処理が実行される。このフィルタ処理は、デジタル回路を用いてデジタル的に実行される。このフィルタ処理を行なうデジタル回路はデジタルフィルタと呼ばれる。

【0004】 デジタルフィルタは、基本的には、加減算器と、1標本化周期（アナログ→デジタル変換されたときのサンプリング信号周期：または入力デジタル信号が与えられる周期）遅延させる遅延素子Dを基本構成要素として備える。このようなデジタルフィルタは、広く一般に用いられるようになってきている。

## 【0005】

【発明が解決しようとする課題】 デジタルフィルタを構成する場合、そのハードウェア構成を固定的に設定したならば、アナログフィルタ回路では可能であったフィルタ係数の変更およびタップ数の変更は容易に行なうことができない。デジタルフィルタにおいては、フィルタ係数は、乗算器の係数で表わされ、この乗算係数の変更は、ハードウェア構成が固定された場合困難となるからである。また、デジタルフィルタにおいては、タップ数は、遅延回路の段数（または乗算器の数）で決定される。タップ数を変更するためには、遅延回路の段数（または乗算器の数）を変更する必要がある、容易には行なうことができない。

【0006】 上述のような固定されたハードウェア構成を備えるデジタルフィルタの欠点を克服する方法として、以下の2種類の方法が考えられる。

【0007】 その1つは、係数（乗算係数）を任意の値に変更することのできる専用LSIを用いることであり、もう1つは、プログラム可能なプロセッサを用いるものである。

【0008】 第1の手法においては、タップ係数（乗算係数すなわちフィルタ係数）を外部から与える構成が用いられる。この場合、タップ係数は変更可能であるものの、デジタルフィルタ自体の構造は固定されている。FIR（有限長インパルス応答）フィルタ、IIR（無限

長インパルス応答) フィルタ、または両者を組合わせたフィルタ構成の場合、他の構成に変更することはできない。そのため、その用途が限定されるという欠点が生じる。

【0009】第2の方法は、DSP (デジタル信号プロセサ) または従来のデータ駆動型プロセサなどの信号処理用プロセサを用いるものである。これらの信号処理用プロセサを用いてデジタルフィルタを構成した場合には、プログラムによりその処理内容を変更することができる。しかしながら、データ処理を行なうためのプログラムには、フィルタ処理そのものの処理を行なう命令以外の命令が含まれる。プロセサを用いてデジタルフィルタを構成した場合、その最小演算命令は、加算、減算、乗算などの命令であり、このため命令の数が増大する。

【0010】データ駆動型プロセサにおいては、あるノードにおいて必要とされる条件が揃ったときにそのノードが発火する。各ノードにはそれぞれ演算命令が割当てられる。したがって、このような単位演算命令を用いてデジタルフィルタを構成した場合、命令の数が増えるため、応じて等価的にノードの数が増大し、発火条件の判別、必要データの待合わせなどフィルタ処理以外のデータ処理を実行する必要があり、このフィルタ処理以外の処理実行時間がオーバーヘッドとなり、高速でフィルタ処理を行なうことができないという問題が生じる。

【0011】上述のように、専用LSIを用いてデジタルフィルタを構成した場合、そのハードウェア構成は固定され、プログラム可能ではないため、任意のフィルタを構成することができないという欠点がある。

【0012】また、この理由により、処理内容としてフィルタ処理を含む複雑なアプリケーションプログラムには、専用LSIで構成されるデジタルフィルタを利用することができないという欠点が生じる。

【0013】またプログラム可能なプロセサを用いた場合にはその命令の内容がフィルタ処理の基本要素となっていないため、命令実行時間によるオーバーヘッドが生じ、高速でフィルタ処理を行なうことができないという問題が生じる。

【0014】それゆえ、この発明の目的は、高速でフィルタ処理をその内容に含むプログラムを実行することのできる情報処理装置を提供することである。

【0015】この発明の他の目的は、任意のフィルタ構造を容易に実現することのできるデジタルフィルタ装置を備える情報処理装置を提供することである。

【0016】

【課題を解決するための手段】この発明に係るデータ駆動型情報処理装置は、複数のレジスタを有するレジスタファイルを備える。このレジスタファイルは、データを格納するデータフィールドと、このデータフィールドにおけるレジスタを特定する行先情報を格納する行先フィールドとを含む。このレジスタファイルにおける1エン

トリーはデータフィールドのレジスタと行先フィールドのレジスタとを含む。好ましくは、このレジスタファイルの1エントリーが、フィルタ処理における基本演算命令に対応する。

【0017】この発明に係るデータ駆動型情報処理装置は、さらに、処理されるべきプログラムに付されたノードに対応して命令を格納するプログラムメモリ手段と、オペランドとノード特定番号とを少なくとも含むデータパケットにตอบสนองしてプログラムメモリ手段へアクセスし、該ノードに対応する命令を読み出す命令読み手段と、乗算器と加算器シフト複合演算器とを含む演算手段と、プログラムメモリ手段から読み出された命令がこの演算手段を用いる演算を指定しているか否かを判別する判別手段と、この判別手段の判別結果が演算手段の利用を示すとき、この命令と、この命令にリンクして読み出されるノード特定番号とを前記データパケットに含まれるオペランドとともにパケットの形態に変換して送出する命令送出手段とを含む。

【0018】この発明に係るデータ駆動型情報処理装置はさらに、命令送出手段から与えられるデータパケット内のオペランドデータをレジスタファイルのデータフィールド内の所定のレジスタへ書き込むデータ入力手段と、命令送出手段からのパケット内の命令に従ってレジスタファイル内のデータフィールドに含まれるレジスタの内容を読み出し、この演算手段へ与えるデータ読み手段と、命令送出手段からのデータパケット内の命令に従って演算手段の出力をレジスタファイルのデータフィールド内のレジスタへ書き込むデータ書き込み手段を含む。このデータ読み手段が読み出すレジスタの指定は、プログラム可能である。また、データ書き込み手段は、行先フィールド内のレジスタのプログラム可能な行先情報が示すレジスタへデータを書き込む手段を含む。

【0019】この発明に係るデータ駆動型フィルタ装置は、さらに命令送出手段からの命令に従ってレジスタファイル内の所定のレジスタの内容を読み出して命令読み手段へデータパケットの形態で伝達するデータ出力手段を備える。

【0020】

【作用】プログラムメモリ手段には、プログラムの各命令がノードに対応して格納され、与えられたデータパケットに従って対応の命令がプログラムメモリ手段から読み出される。この読み出された命令が演算装置を利用する命令の場合には、この読み出された命令に従ってレジスタファイルおよび演算手段のデータの書き込みおよび読み出しが実行される。

【0021】データ駆動方式で命令が実行されるため、高速でプログラムを実行することができる。また、この命令によりフィルタ構造を定義することができるため、フィルタ処理をその内容に含むプログラムであっても高速に実行することができる。

【0022】演算手段は、フィルタ処理に必要とされる乗算器と加算シフト複合演算器を含む。この演算手段は、デジタルフィルタの基本ユニットを構成する。デジタルフィルタのデータフィールド内のレジスタの内容は演算手段へ与えられ、所定の演算が実行される。この演算手段による演算はフィルタの基本単位の演算である。したがって、このデータレジスタの読出／書込命令は、フィルタ処理の基本単位命令となっており、フィルタ処理以外の命令を実行する必要がなく高速でフィルタ処理を行なうことができる。

【0023】またデータレジスタの書込／読出先を変更することによりFIRフィルタおよびIIRフィルタいずれをも実現することができるのみならず、所望の構造のフィルタを構築することができる。

【0024】また、乗算係数（フィルタ係数）をデータレジスタ内に保持しておくことにより容易にフィルタ係数を変更することができ、フィルタ特性の変更を容易に行なうことができる。

【0025】また、レジスタファイルのエントリーの数を調整することにより容易にフィルタのタップ数を変更することができる。

【0026】

【実施例】図2はこの発明において利用されるデータ駆動型フィルタ装置の全体の構成を概略的に示すブロック図である。図2において、データ駆動型フィルタ装置は、入力データを受ける入力部20と、複数のレジスタを含み、処理されるべきデータおよび処理後のデータを格納するレジスタファイル10と、このレジスタファイル10に格納された最終演算結果を読出して装置外部へ出力する出力部30を含む。

【0027】レジスタファイル10は、その構成は後に詳細に説明するが、データ駆動型構成を実現するために、処理データのみならず、この処理データの書込先を特定する行先情報を格納するレジスタも備える。すなわち、レジスタファイル10は、処理データを格納するデータフィールドと、このデータフィールド内の特定のレジスタを特定する行先情報を格納する行先フィールドとを含む。このデータフィールドおよび行先フィールドはそれぞれ複数のレジスタを含む。

【0028】入力部20は、レジスタファイル10の所定のレジスタ（この所定のレジスタはプログラムにより指定され、実現すべきフィルタ構造によって異なる）へ入力データを書込む。入力部20はまた、後に説明するが、1つの入力データに関連する処理が完了するまで次のデータの入力を待合わせる機能を備える。

【0029】出力部30は、このレジスタファイル10の所定のレジスタ（フィルタの最終ユニットに対応するレジスタ）の内容を読出して装置外部へ出力する。この構成についても後に詳細に説明する。

【0030】データ駆動型フィルタ装置はさらに、乗算

器1a～1nを含む第1の演算器群40と、加算シフト複合演算器2a～2nを含む第2の演算器群50を含む。この第1の演算器群40および第2の演算器群50に含まれる乗算器および加算シフト複合演算器の数は複数ではなく、1つであってもよい。1個の乗算器1（乗算器1a～1nを総称的に示す）と1個の加算シフト複合演算器2（加算シフト複合演算器2a～2nを総称的に示す）で1つの単位フィルタ構造が実現される。この第1の演算器群40および第2の演算器群50へは、レジスタファイル10のデータレジスタの内容が伝達される。処理結果は、この第1の演算器群40および第2の演算器群50へ与えられたデータと同一エントリーの行先レジスタの内容が指定するレジスタへ書込まれる。特に、第1の演算器群40の処理結果が書込まれるレジスタは予め固定的に設定される。

【0031】レジスタファイル10は、単一代入単一読出のルールに従ってデータの書込および読出が実行される。ここで「単一代入単一読出」のルールとは、一度読出されたデータは消費され、再び利用されることがないというルールである。また、このルールでは書込みは読出しが完了するまで行なわれない。ただしこのルールは、定数データ（フィルタ係数：乗算係数）は除く。定数データは、オーバーライトされたときには、新しいデータが有効となる。

【0032】図3は、図2に示すレジスタファイルの内部構成を具体的に示す図である。図3において、レジスタファイル10は、情報を格納するためのレジスタファイルRFと、このレジスタファイルRFのデータ書込および読出を制御する書込／読出制御部RWCを含む。レジスタファイルRFは、データフィールドDAFと、行先フィールドDEFを含む。データフィールドDAFは、複数のデータレジスタDARを含み、行先フィールドDEFは、複数の行先レジスタDERを含む。

【0033】図3において同一行に配置されるレジスタがこのレジスタファイルの1エントリーを構成する。データレジスタDARのデータが読出されて第1または第2の演算器群へ与えられたとき、同様に関連の行先レジスタDERの内容が読出され書込／読出制御部RWCへ与えられる。書込／読出制御部RWCは、第1または第2の演算器群からの出力データをこの行先レジスタDERからの行先情報に従って対応のデータレジスタへ書込む。書込／読出制御部RWCはデータレジスタDARおよび行先レジスタDERそれぞれ独立にその書込／読出を制御することができる。

【0034】レジスタファイル10はさらに、この書込／読出制御部RWCの動作を制御するプログラムコントロール部100を含む。プログラムコントロール部100は、このフィルタ装置が実現するフィルタ処理内容を決定するプログラムを格納するプログラムメモリPMと、このプログラムメモリPMに格納されたプログラム

に従って書込／読出制御部RWCの動作を制御するシーケンスコントローラSCを含む。この書込／読出制御部RWCは、したがってシーケンスコントローラSCの出力情報に従って、順次、または並列にレジスタファイルRFの内容の書込および読出、ならびにバス接続制御を実行する。

【0035】図3においては、入力部20および出力部30はともに書込／読出制御部RWCを介してデータの入出力を行なうように示されている。これは書込／読出制御部RWCが入出力インタフェース回路の機能をも併せて備えるためである。さらに図3においては、入力部20および出力部30はシーケンスコントローラSCの制御を受けるように示される。これはプログラムメモリPMに書込まれたプログラムに従って入力部20が入力データINを書込むデータレジスタを特定するアドレスおよび出力部30が出力データOUTを生成するためのデータフィールドDAFの特定のデータレジスタDARを指定するアドレスがそれぞれこのシーケンスコントローラSCの制御の下に設定されることを示すためである。

【0036】シーケンスコントローラSCが入力部20および出力部30にこのレジスタファイルRF内のデータレジスタのアドレスを設定することではなく、書込／読出制御部RWCがシーケンスコントローラSCの制御の下に入力部20および出力部30とデータの入出力を行なう構成とされてもよい。このとき、書込／読出制御部RWCはシーケンスコントローラSCの制御の下に入力データINの書込および出力データOUTの読出をレジスタファイルRFに対して実行する。

【0037】プログラムメモリPMへは、外部からプログラムを書込むことができる（この書込経路は示さず）。これにより、適用用途に応じたフィルタ構造を実現することができる。

【0038】また図3においては、レジスタファイル10内部にプログラムコントローラ100が設けられるように示されている。このプログラムコントローラ100は、レジスタファイル10の外部に設けられて装置外部から書込／読出制御部RWCを制御する構成が用いられてもよい。

【0039】[レジスタの構成] 図4は、レジスタファイルの構成および接続形態を示す図である。図4において、レジスタファイル10は、1エンタリーとしてデータレジスタDARと行先レジスタDERを含む。

【0040】このデータレジスタDARのレジスタの配置は、命令iirおよび命令firに従ってその意味が変更される。命令iirは、IIRフィルタ処理を実行するための命令であり、命令firはFIRフィルタ処理を実行するための命令である。図4においては、この命令iirが指定されたときのレジスタの配置を示す。

【0041】図4において、データレジスタDARは、

処理されるべきデータを格納するA1レジスタ、A2レジスタ、Cレジスタ、入力データを格納するBレジスタ、乗算結果(Xの計算結果)を格納するXレジスタ、処理データを格納するDレジスタを含む。

【0042】A1レジスタおよびCレジスタに格納されたデータは、第1の演算器群40に含まれる乗算器1へ与えられる。乗算器1の乗算結果(Xの計算結果)は書込回路62を介して同じエンタリー内のXレジスタへ格納される。

10 【0043】Bレジスタ、XレジスタおよびDレジスタの格納するデータは第2の演算器群50に含まれる加算シフト複合演算器2へ与えられる。加算シフト複合演算器2は、加算器22と、加算器22の出力を所定ビットシフトするシフト回路24を含む。BレジスタおよびXレジスタの格納するデータが加算器22へ与えられる。Dレジスタの格納するデータはシフト回路24におけるシフトビット数を決定する。シフト回路24は、そのシフト数がプログラマブルな回路であり、そのシフト操作により除算を実行する。

20 【0044】行先レジスタDERは、加算シフト複合演算器2の出力を格納する位置を示す行先情報を格納するEレジスタと、A1レジスタのデータを書込むレジスタを特定する行先情報を格納するFレジスタを含む。このEレジスタおよびFレジスタはそれぞれ複数の行先を特定することができる。EレジスタおよびFレジスタの格納する行先情報はプログラム可能であり、プログラムメモリPM(図3参照)に格納された命令に従ってシーケンスコントローラSCおよび書込／読出制御部RWCの制御の下に対応の行先情報がプログラムされる。

30 【0045】図4においては、さらにこのデータの書込および読出を実行するために書込回路61、62および63が示される。書込回路61は、Fレジスタに格納された行先情報をデコードし、対応のレジスタへA1レジスタの格納するデータを書込む。書込回路62は乗算器1の出力を同一エンタリー内のXレジスタへ書込む。書込回路63は、Eレジスタに格納された行先情報をデコードし、加算シフト複合演算器2の出力を対応のレジスタへ書込む。この書込回路61、62および63は図3に示す読出／制御部RWCに含まれる。また、このデータレジスタDARおよび行先レジスタDERからのデータ読出は、それぞれ独立に、図3に示す読出／書込制御部RWCにより実行される。各レジスタは読出／書込制御部RWCの制御の下にそれぞれ対応のデータバスに接続される。このレジスタとデータバスとの接続はプログラムにより決定される。

40 【0046】上述のようにレジスタファイルにおける1エンタリーのレジスタにおいて、各レジスタはどの演算に利用されるかはプログラムにより固定的に設定される。すなわちAレジスタおよびCレジスタは乗算用のデータを格納するために利用され、BレジスタおよびDレ

レジスタは加算シフト複合演算のために利用される。それら以外の演算には利用されない。また、各レジスタには、初期値および読出専用の定数を格納することができる。このレジスタからのデータの書込および読出は単一代入／単一読出（1回限りの書込および読出）のルールに従って実行される。定数が格納されたレジスタは、書込は行なわれず、その定数を複数回読出すことができる。

【0047】また、行先レジスタDERに含まれるEレジスタおよびFレジスタの行先情報はプログラム可能である。したがって、任意の構造のフィルタを容易に構成することができる。

【0048】図5はf i r命令によるレジスタの接続構成を示す図である。命令f i rの場合、データレジスタDARは、Aレジスタ、Cレジスタ、Bレジスタ、Xレジスタ、およびDレジスタを含む。CレジスタおよびBレジスタの格納するデータは乗算器1へ与えられる。すなわち、CレジスタおよびBレジスタは乗算のためにのみ利用されるデータを格納する。Aレジスタ、XレジスタおよびDレジスタに格納されるデータは、加算シフト複合演算器2へ与えられる。Xレジスタへは、この乗算器1の乗算結果が書込回路62を介して書込まれる。

【0049】上述のように、f i r命令においても、各データレジスタは予め定められた演算のためにのみ利用される。

【0050】行先レジスタDERは、Bレジスタに格納されたデータを書込むレジスタを特定する行先情報を格納し、Fレジスタは加算シフト複合演算器2の出力データを格納するレジスタを特定する行先情報を格納する。

【0051】このf i r命令においても、各レジスタには初期値および読出専用の定数を置くことができる。各レジスタ（データレジスタ）に対しては単一代入／単一読出のルールに従ったデータの書込および読出が行なわれる（読出専用の定数を除く）。

【0052】EレジスタおよびFレジスタに格納されるデータはプログラム可能である。したがって、このf i r命令の場合においてもフィルタの構成を容易に変更することができる。

【0053】この図4および図5に示すレジスタはレジスタファイルにおける1エントリーを構成する。1エントリーは、1つのi i r命令またはf i r命令に対応する。命令により各レジスタと乗算器および加算シフト複合演算器との間の接続が決定される。

【0054】図6は、i i r命令およびf i r命令両者を実現するためのレジスタファイルの1エントリーの構成およびレジスタ間の接続を示す図である。図6において、レジスタファイルは図4に示す構成と同様、データレジスタDARがA1レジスタ、A2レジスタ、Cレジスタ、Bレジスタ、Xレジスタ、およびDレジスタを含む。行先レジスタDERはEレジスタおよびFレジスタ

を含む。レジスタA1、A2、C、B、X、D、EおよびFは1エントリーを構成する。

【0055】命令f i rおよび命令i i rに従って演算器群40および50へ与えられるデータを切換えるためにスイッチ回路71、72および73が設けられる。スイッチ回路71は、命令f i rおよびi i rに従ってA1レジスタおよびBレジスタの伝達経路を切換える。すなわち、スイッチ回路71は、f i r命令の場合にはBレジスタのデータを左出力Lへ出力し、A1レジスタの格納データを右出力Rへ伝達する。スイッチ回路71は、i i r命令の場合には、Bレジスタの格納データを右出力Rへ伝達し、A1レジスタの格納データを左出力Lへ伝達する。

【0056】スイッチ回路72は、f i r命令の場合にはスイッチ回路71の左出力Lから与えられたデータを左出力Lへ伝達し、i i r命令の場合には与えられたデータを右出力Rへ伝達する。スイッチ回路73はスイッチ回路72と同様の動作を行ない、f i r命令の場合には入力データを左出力Lへ伝達し、i i r命令の場合には入力データを右出力Rへ伝達する。

【0057】図6においては、スイッチ回路72および73へ行先レジスタDERに含まれるEレジスタおよびFレジスタの格納データが伝達されるように示されている。すなわち、このスイッチ回路72および73が行先情報をデコードし、データ伝達線を振分ける機能を備えるように示される（図4および図5に示す書込回路の機能を備える）。これは、スイッチ回路72および73は単にデータ伝達経路を切換える機能のみを備え、そのスイッチ回路の出力に図4および図5に示す書込回路61、および63が設けられる構成が利用されてもよい。

【0058】この図6に示すようにスイッチ回路71、72および73のデータ伝達経路を命令i i rおよびf i rに従って切換えることにより、同一のレジスタ構成を用いてf i r命令およびi i r命令いずれをも実行することができ、FIRフィルタおよびIIRフィルタいずれをも実現することができ、かつまた両者を組合わせた結合型フィルタをも構成することができる。

【0059】[命令ブロック構造] i i r命令およびf i r命令はデータフロー形式で表現される。以下、この命令の記述および内部構造について説明する。

【0060】図7はi i r命令の記述および内部構造を示す図である。図7(A)において、i i r命令は4入力2出力命令である。入力A、B、CおよびDは、図4に示すレジスタに対応し、また出力EおよびFはレジスタEおよびFに対応する。入力データとしては、初期値または定数が含まれてもよい。

【0061】図7(B)において、i i r命令は、4つのノード81、82、83および84を含む。ノード81は加算ノードであり、ノード82は乗算を行なうノードである。ノード83はデータを所定ビットシフトさせ



るシフト演算を行なうノードであり、ノード84は与えられたデータ所定時間遅延させる遅延 ( $Z^{-1}$ ) を実行するノードである。

【0062】ノード81は、乗算ノード (Xノード) の出力とB入力に与えられたデータを加算する。ノード82は、C入力とノード84の出力またはA1入力とを乗算する。ノード83は、ノード81 (加算ノード: +ノード) の出力をD入力に与えられたビット数だけシフトさせる。ノード84はA2入力を所定時間 ( $1T: T$  は1サンプリング期間) 遅延させる。入力A1およびA2はどちらか一方が入力される。同時に入力A1およびA2が与えられることは禁止される。ノード84の出力がF出力 (Fレジスタの行先情報が特定するレジスタ) へ伝達され、ノード83の出力がE出力 (Eレジスタの行先情報が特定するレジスタ) へ伝達される。

【0063】図4に示す構成においては、ノード84の実行する遅延 ( $Z^{-1}$ ) を実現する演算器は示されていない。これは等価的にデータレジスタDARにおけるA2レジスタからA1レジスタへのデータ転送により実現される (図4においてA2レジスタからA1レジスタへの矢印で示す)。

【0064】図8は、f i r命令の記述および構造を示す図である。図8 (A) において、f i r命令は、4入力2出力の命令で表現される。入力A、B、CおよびDは、図5に示すAレジスタ、Bレジスタ、CレジスタおよびDレジスタに対応し、E出力およびF出力はEレジスタおよびFレジスタの格納する行先情報が特定する行先へ結合されることを示す。

【0065】図8 (B) において、f i r命令は、その内部構造として、乗算 (X) を行なうノード91と、加算 (+) を行なうノード92と、所定ビットだけ入力データをシフトさせる操作を行なうノード93と、ノード93の出力を所定時間遅延させる遅延操作 ( $Z^{-1}$ ) を行なうノード94を含む。ノード91は、入力BおよびCの乗算を行なう。ノード92は、乗算器91の出力と入力Aとの加算を行なう。ノード93は、加算ノード92の出力をD入力が指定するビット数シフトさせて出力する。ノード94はノード93の出力を1サンプリング期間遅延させて出力する。

【0066】図7 (B) および8 (B) に示す加算ノード (81, 92) およびシフト操作ノード (83および93) は図2に示すように加算シフト複合演算器として1つの演算器で表現される。

【0067】図7および図8に示すi i r命令およびf i r命令の内部構造はそれぞれI I RフィルタおよびF I Rフィルタの単位構造に対応する。次に一般的なフィルタをこの命令i i rおよびf i rが示す内部構造へ分解する方法について説明する。

【0068】[フィルタのプログラム実施例] まずデジタルフィルタの構成の基本について説明する。

【0069】標本化周期Tで標本化された入力系列を  $\{x_n\} = x_0, x_1, x_2, \dots, x_n, \dots$  とし、フィルタの出力系列を  $\{y_n\} = y_0, y_1, y_2, \dots, y_n, \dots$  とする。一般に、このデジタルフィルタにおいては入力系列  $\{x_n\}$  と出力系列  $\{y_n\}$  とは差分方程式を用いて以下のように関係付けられる。

【0070】 $y_n = \sum a_k \cdot x_{n-k} - \sum b_k \cdot y_{n-k}$   
ここで、第1項の総和は  $k=0 \sim M$  に対して行なわれ、第2項の総和は  $k=1 \sim N$  に対して実行される。

【0071】すなわち、時刻  $t = nT$  におけるフィルタの出力  $y_n$  は、入力  $x_{n-M}$  から  $x_n$  の  $(M+1)$  個の過去および現在の入力と、出力  $y_{n-N}$  から  $y_{n-1}$  までの  $N$  個の過去のフィードバックされた出力 (係数  $b_k$  が0でない場合) との差分により決定される。

【0072】通常この入力系列  $\{x_n\}$  および出力系列  $\{y_n\}$  は  $z$  変換される。 $z$  変換により、標本値入力系列  $\{x_n\}$  は、

$$X(z) = \sum x_n \cdot z^{-n}$$

と表現される。総和は  $n=0$  から無限大に対して実行される。 $z^{-1}$  は単位遅延演算子と呼ばれる。この  $z$  変換により、出力  $Y(z)$  は、

$$Y(z) = H(z) \cdot X(z)$$

と表現される。 $H(z)$  は伝達関数と呼ばれる。上述の式において、係数  $b_k$  がすべて0の場合には、出力信号  $y_n$  は、過去の出力信号を用いないで決定される。このフィルタ処理を実現するデジタルフィルタは非巡回型デジタルフィルタと呼ばれ、トランスバースルフィルタ (遅延素子の各タップ出力の加重和を得る構成) が一般に古くから知られている。

【0073】一方、係数  $b_k$  のうち少なくとも1個0でない係数が存在する場合、すなわち過去の出力が現在の出力に影響を及ぼす場合この構成は巡回型デジタルフィルタと呼ばれる。巡回型フィルタの伝達関数を指数に展開すると、通常は  $z^{-1}$  の無限級数で表現される。すなわちインパルス入力に対して出力が無限に続く。このためこの巡回型フィルタを無限インパルス応答フィルタ (I I Rフィルタ) と呼ぶ。一方、非巡回型フィルタでは、この伝達関数の級数は有限である。このため非巡回型フィルタは有限インパルス応答フィルタ (F I Rフィルタ) と呼ばれる。

【0074】まず、代表的なフィルタの構成について説明し、次にこれを図7に示す命令構造に分解する手順について説明する。まず2次のI I Rフィルタを構成する場合について説明する。2次のI I Rフィルタは、遅延が2次項までであるフィルタを示す。

【0075】図9は2次I I Rフィルタの構成と命令への分解手順を示す図である。図9において、2次I I Rフィルタは、縦列接続された加算器101、102、103、および104と、加算器104の出力を係数4で除算する除算器105を含む。I I Rフィルタはさら



13

に、加算器102の出力を1サンプリング周期遅延させる遅延回路105と、遅延回路105の出力に所定の係数0.6を乗算して加算器102へ与える乗算器106と、遅延回路105の出力に係数0.7を乗算して加算器103へ与える乗算器107と、遅延回路105の出力をさらに1サンプリング期間遅延させる遅延回路108と、遅延回路108の出力に係数0.5を乗算して加算器101へ与える乗算器109と、遅延回路108の出力に係数0.8を乗算して加算器104へ与える乗算器110を含む。

【0076】この図9に示す2次IIRフィルタを、次に図7に示すiir命令に分解する。iir命令は、加算(+)操作、シフト操作、乗算操作(X)、および遅延操作を含む。シフト操作により除算を実行することができる。この図9に示す2次IIRフィルタを加算、シフト、乗算および遅延の単位に分割する。

【0077】すなわち、加算器101、遅延回路108、および乗算器109を、第1のiir命令で表現する。加算器102、遅延回路105、および乗算器106を、第2のiir命令で表現する。加算器103および乗算器107を第3のiir命令で表現する。加算器104、除算器105および乗算器110を第4のiir命令で表現する。この第1ないし第4のiir命令#1~#4をフローグラフで表わすと図10に示す記述が得られる。

【0078】乗算器106、107、109および110の乗算係数はそれぞれの命令におけるC入力へ与えられる定数により表現される。

【0079】図10において、IIRフィルタは、4つの縦列接続されたiir命令#1~#4により表現される。第1のiir命令#1は、A入力に、第2のiir命令#2のF出力を受け、B入力に外部からの入力データを受け、C入力に定数0.5を受け、D入力に定数0を受ける。第1のiir命令#1のA入力には図7に示すA2入力を利用される。

【0080】第2のiir命令#2は、自身のF出力をA入力に受け、第1のiir命令#1のE出力をそのB入力に受け、C入力に定数0.6を受け、D入力に定数0を受ける。第2のiir命令#2はそのA入力として、A2入力を利用する。

【0081】第3のiir命令#3は、そのA入力に第2のiir命令#2のF出力を受け、B入力に第2のiir命令#2のE出力を受け、そのC入力に定数0.7を受け、D入力に定数0を受ける。第3のiir命令#3は、そのA入力としてA1入力を利用する。

【0082】第4のiir命令#4は、そのA入力に、第1のiir命令のF出力を受け、そのB入力に、第3のiir命令#3のE出力を受け、そのC入力に定数0.8を受け、D入力に定数2を受ける。第4のiir命令#4のE出力からフィルタ処理結果が出力される。

14

【0083】iir命令のD入力には、シフト操作におけるシフトビット数を決定するデータが与えられる。このシフト操作によりデータの割算が実現される。図9に示すフィルタ構成においては、除算操作は除算器105においてのみ実行されている。この係数4による除算は、2ビット下位方向へデータをシフトさせることにより実現される。したがって、第4のiir命令#4のD入力に対して定数2が与えられる。残りのiir命令#1~#3に対しては除算操作は実行されないため、それぞれのD入力へは定数0が与えられる。

【0084】図11は、図10に示すフローグラフをテキスト形式で記述したプログラムを示す図である。図11において右端の番号1~6はプログラム番号を示し、左端の番号0~4はレジスタファイルにおける行番号(エントリー番号)を示す。この行番号はデータの行先を示す行の番号と1対1に対応する。次にプログラムの内容について説明する。

【0085】(1) プログラム番号1の「initial」は初期値の宣言を示す。「1(A2)=0」は、第1のiir命令#1のA2レジスタの格納値、すなわち遅延操作 $Z^{-1}$ の入力初期値が0であることを示している。

【0086】また「2(A2)=0」は、第2のiir命令#2のA2レジスタの初期値が0であることすなわちその操作( $Z^{-1}$ )の初期値が0であることを示す。

【0087】(2) プログラム2の「input」は外部からのデータ入力を示す。「1(B)」はこの入力データを第1のiir命令#1に対応する第1行のレジスタBに格納することを示す。

【0088】(3) プログラム番号3の「iir」はiir命令を示す。iir(, , 0.5, 0)はiir命令に利用される定数を示す。括弧内の値は、各入力の定数すなわちiir(A2, B, C, D)の対応の入力に対する定数を表わす。プログラム番号3のプログラムにおいては以下のことが示される。

【0089】A2入力に対しては定数値は設定されない(変数のみ)。B入力に対しても定数値は設定されない。C入力に対しては定数0.5が設定される。D入力に対しては定数0が設定される。定数値が存在しない場合は「,」のみで示す。すなわち、第1のiir命令(プログラム番号1)に対しては、乗算係数として0.5が設定され、またシフトビット数としては0が設定されシフト動作が行なわれず、除算が行なわれないことを示す。

【0090】「2(B)!4(H)」はこのiir命令のデータの行先を示し、括弧の前に付された番号が行先行数すなわち、E出力およびF出力の行先を示しており、このE出力とF出力の行先は!により区別される。プログラム番号3においては以下のことが示される。

【0091】2(B):E出力は2行目の第2のiir

命令のB入力へ与えられる。F出力は第4行の第4のi i r命令のA1入力へ与えられる。

【0092】(4) プログラム番号4についてはi i r命令の各定数はプログラム番号3のi i r命令のそれらと同様である。データ出力先として複数個存在する。すなわち、E出力は第2行のA2入力へ与えられかつ第3行のi i r命令のB入力へ与えられる。第1ないし第4のi i r命令はレジスタファイルにおける各行と対応しており、したがって、レジスタファイルの第2行および第3行のA2レジスタおよびBレジスタへこの第2のi i r命令のE出力が伝達されることを示す。

【0093】F出力は第1のi i r命令のA2入力および第3のi i r命令のA1入力へ伝達される。すなわちレジスタファイルにおける第1行および第3行のA2レジスタおよびA1レジスタへこの第2のi i r命令のF出力が入力される。

【0094】(5) プログラム番号5のi i r命令についてはその定数およびE出力の定義については上で説明したものと同一である。「absorb」は、行先がないことを示す。すなわち第3のi i r命令のF出力は出力されないことを示す。

【0095】(6) プログラム番号6においてi i r命令の定義およびF出力の行先については上の第3のi i r命令の場合と同様である。「output」はこの第4のi i r命令のE出力が出力データとして装置外部へ出力されてフィルタ処理結果を示すデータとして利用されることを示す。

【0096】このテキスト形式で記述されたプログラムはレジスタファイルの各行にそれぞれ対応する。すなわち、第1ないし第4のi i r命令はそれぞれレジスタファイルの第1行ないし第4行のエントリに対応する。このプログラムに従ってレジスタファイルへのデータの書込および読出が実行される。

【0097】[フィルタ装置の具体的動作]次に具体的な動作を以下の順序で説明する。

【0098】(a) 動作の規則を説明する。(b) レジスタの状態を実行順序に従って示す。(c) 実行効率を調べ、無駄なオーバーヘッドが生じていないことを実証する。および(d) 複数(4つ)の演算器を持った場合の動作を説明する。

【0099】「基本ルール」

(1) A2レジスタ、A1レジスタおよびBレジスタには初期値が格納される。これらのレジスタA2、A1およびBの格納データは新しく入力されたデータにより更新される。CレジスタおよびDレジスタに格納される定数は更新されない。定数でない場合には新しく与えられる入力データにより更新される。定数でないか否かはi i r命令により決定される。

【0100】(2) すべてのレジスタの値は1回書込1回読出に限定される。すなわち単一代入単一読出のル

ールが適用される。ただし定数を除く。

【0101】(3) レジスタA1に格納されたデータと加算シフトを含む演算器の計算結果を複数かつ任意の行へ書込むことができる(プログラム可能である)。この行先はそれぞれE出力およびF出力としてプログラム(図11参照)時において各命令に対応してプログラムにより指定される。

【0102】(4) 加算シフト複合演算を除く演算の結果は予め定められたレジスタに書込まれる(Xレジスタ)。

【0103】以下に説明する計算の実施例においては第1の演算器群40は1個の乗算器を含み、第2の演算器群50は1個の加算シフト複合演算器を含むと想定する。

【0104】「計算のルール」

(1) 演算は、行番号すなわちプログラム番号にかかわらずデータが揃ったものから順次実行される。

【0105】(2) 演算は、加算シフト、次いで乗算の優先順位で実行される。

(3) 同じ演算に対するデータが同時に揃った場合には行番号の逆順にその演算が実行される。すなわち行番号の大きい方の命令に従った演算が先に実行される。

【0106】(4) A2レジスタからA1レジスタへのデータの移動(遅延の表現)およびF出力におけるデータのコピー(Fレジスタが複数の行先情報を含む場合)は、すべて演算の前に実行される。

【0107】(5) 行先が複数存在する場合には、行先がすべて空いていない状態でない限りその演算は実行されない。すなわち演算は、行先が空いておりかつデータが揃った場合に実行される。以下の説明においては加算シフト複合演算のE出力および乗算操作結果のF出力による転送の場合にこのルールが適用される。

【0108】(6) 後の演算が前の演算より優先されることはない(演算順序における追越しは生じない)。

【0109】次にこの図9に示す2次IIRフィルタ処理動作を実行する際のレジスタファイル内のデータの動きについて説明する。

【0110】図12は、レジスタの初期状態と各レジスタと演算器との関係を示す図である。レジスタの初期状態は図11に示すプログラムが書込まれ、内部状態が初期設定された状態を示す。この状態において、CレジスタおよびDレジスタにはそれぞれ定数が書込まれる。A1レジスタおよびCレジスタは乗算器に結合される。BレジスタおよびXレジスタは加算器へ結合され、Dレジスタはシフト回路へ結合される。A1レジスタから読出されたデータはFレジスタに格納された行先情報に従って対応のレジスタへ書込まれる。

【0111】シフト回路すなわち加算シフト複合演算器の出力はEレジスタに格納された行先情報に従って対応のレジスタに書込まれる。この図12において左端に示

す番号はレジスタファイルにおける行番号を示しかつ図11に示す各プログラムの行番号と1対1に対応する。1行ないし4行のE1レジスタにはデータ1、0、0、および1が初期値として格納されている。この格納データは単なる一例であり、フィルタ処理動作実行時にはこのA1レジスタに初期設定された値は無効データとして処理される。

【0112】図13は、第1サイクルの動作を示す図である。図13(A)において、第1行のBレジスタへデータ“1”が格納される。この入力動作と並行して、第4行のA1レジスタのデータ“1”を用いた乗算が行なわれ、その乗算結果がXレジスタへ格納される。

【0113】すなわち、図13(B)に示すようにブロック#1の加算器の入力にデータ“1”が与えられ、またブロック#4の加算器の入力にデータ“0.8”が与えられる。

【0114】図14は、第2サイクルの実行動作時におけるレジスタの内容およびそのときのデジタルフィルタにおけるデータの分布状態を示す図である。図14

(A)において、入力部からは新たにデータは与えられず、入力待ち状態となる。第4行のエントリにおいては、BレジスタとXレジスタにデータが揃ったため、加算シフト動作が実行される。第4行に対応する命令の結果はEレジスタの内容で決定されるレジスタに書込まれる。第4のiir命令のEレジスタには「出力」が指定される。したがって、データ“0.2”が外部に出力される。

【0115】一方、第3行のエントリ（命令#3に対応）においては、A1レジスタのデータ“0”とCレジスタのデータ0.7との乗算が実行され、その乗算結果が対応のXレジスタに格納される。

【0116】ここで、図14(A)においてレジスタにおいて空欄の部分はその格納データが消費されたことを示す。単一代入単一読出のルールに従ってレジスタファイルへのデータの書込および読出が実行されているため、書込まれかつ読出されたデータは再び利用されないからである。以下の説明においても同様とする。

【0117】この図14(A)に示すデータの状態をデジタルフィルタにおけるデータに対応付けると図14

(B)に示すようになる。すなわち、ブロック#4において、加算器において0と0.8とが加算され次いでシフト操作による除算が行なわれ、0.2が出力される。

【0118】ブロック#3においては、その乗算器において係数0.7と0との乗算が行なわれ、その乗算結果“0”が加算器へ与えられる。乗算操作と加算シフト複合操作は互いに異なる演算器により実行されるため、互いに並列に実行することができる。

【0119】図15は、第3実行サイクルにおけるレジスタの内容およびフィルタにおけるデータの配置を示す図である。図15(A)に示すように、第3のiir命令

における加算シフト操作結果が第4行のDレジスタへ書込まれ、一方第2行のiir命令のA1レジスタとCレジスタのデータの乗算が行なわれその乗算結果が第2行のXレジスタに格納される。

【0120】すなわち図15(B)に示すように、ブロック#2の加算器の入力“0”が確定状態となり、またブロック#4の加算器の入力、すなわちプログラム#3の加算器出力が“0”に確定する。

【0121】図16は、第4実行サイクルにおけるデータレジスタの内容とそのときのデジタルフィルタにおける確定データの分布を示す図である。図16(A)において、第1行のA1レジスタおよびCレジスタに格納されたデータの乗算が行なわれその乗算結果がXレジスタに格納される。すなわち、図16(B)に示すように、ブロック#1における乗算器の乗算結果が確定し、ブロック#1の加算器の入力が確定する。次いで、遅延素子の操作を実現するためにデータのコピーが実行される。

【0122】図16(A)において、第2行のBレジスタおよびXレジスタのデータは確定している。したがって、第1行のA1レジスタとCレジスタの乗算と並行して加算シフト操作を実行できるように思われる。しかしながらこの場合、計算ルールにおいて後からの演算は前の演算よりも優先されないというルールに従って、この第2行のBレジスタとXレジスタのデータの加算およびシフト動作は行なわれない。

【0123】図17は、遅延操作を実現する際のレジスタのデータの動きを示す図である。図17(A)において、まず第1行のA2レジスタのデータ“0”が第1行および第4行のA1レジスタへコピーされる。これにより、ブロック#1に含まれる遅延素子の操作が実現される。

【0124】図17(B)において、次いで第2行のA2レジスタのデータが第1行のA2レジスタと、第2行および第3行のA1レジスタへコピーされる。これによりブロック#2における遅延素子の遅延操作が実現される。デジタルフィルタにおける各乗算器の入力が確定する。

【0125】図17(C)において、第2行のBレジスタおよびXレジスタのデータに対し加算シフト演算が実行され、この演算結果は第2行のA2レジスタおよび第3行のBレジスタへ格納される。これは、ブロック#2において加算が行なわれ、その加算結果が遅延素子入力およびブロック#3の加算器入力へ与えられて確定状態となったことを示す。

【0126】図18は、図17に示す動作に続く操作におけるデータレジスタにおけるデータの配置を示す図である。図18(A)において、第4行のA1レジスタおよびCレジスタのデータの乗算が行なわれ、その乗算結果がXレジスタに格納される。これはブロック#4における乗算器の出力が確定した状態に対応する。

19

【0127】図18(B)に示すように、上述のコピー動作、加算およびシフト演算操作、および乗算操作を順次実行すると、最終的にデータとしては、第3行のBレジスタおよび第4行のXレジスタのデータが確定した状態となる。すなわち、上述の操作を実行することにより図18(C)に示すごとくブロック#3における加算器の入力およびブロック#4における乗算器出力が確定した状態となる。この上述のコピー動作は常に遅延操作を実行する場合行なわれる。以下の説明においてはこの図17(A)ないし図18(A)に示す操作は省略し、図18(B)に示す最終状態のみを示す。

【0128】図19は、第6実行サイクルにおけるデータのデータレジスタにおける配置およびデジタルフィルタにおける確定データの状態を示す図である。図19

(A)に示すように、第3行のA1レジスタおよびCレジスタの乗算が行なわれその乗算結果が第3行のXレジスタに書込まれる。この乗算操作と並行して、第4行のBレジスタおよびXレジスタを用いた加算シフト演算操作が実行され、その演算結果のデータ“0”が出力データとして出力される。

【0129】図20は、第7実行サイクル時におけるレジスタファイルにおけるデータの配置およびデジタルフィルタにおける対応のデータの配置状態を示す図である。図20(A)に示すように、第2行のA1レジスタとCレジスタのデータを用いた乗算が行なわれ、該乗算結果がXレジスタに格納される。この乗算操作と並行して、第3行のBレジスタとXレジスタの加算およびシフト演算操作が実行され、この演算操作結果が第4行のBレジスタに格納される。

【0130】すなわち、図20(B)に示すように、ブロック#2における加算器の入力すなわち乗算器の出力が確定しかつブロック#4の加算器の入力すなわちブロック#3の出力が確定状態となる。

【0131】図21は第8実行サイクルにおけるデータレジスタにおけるデータの配置およびそのときのデジタルフィルタにおけるデータの配置形態を示す図である。図21(A)において次の入力データ“2”が第1行のBレジスタへ格納される。この格納と並行して、第1行のA1レジスタとCレジスタとの乗算が実行され、その乗算結果が第1行のXレジスタへ書込まれる。また同時に第1行のBレジスタとXレジスタのデータに対する加算シフト操作が実行され、その演算操作結果が第2行のBレジスタへ格納される。すなわち、第1行のデータレジスタに対し乗算操作と加算およびシフト演算操作が行なわれかつ新しい入力データの格納が実行される。

【0132】すなわち、図21(B)に示すようにブロック#1の加算器の出力すなわちブロック#2の加算器の入力が確定しかつブロック#1の乗算器の出力が確定する。以降、この新しい入力データ“2”を用いて同様の動作が繰り返される。

20

【0133】図22は、第9実行サイクルにおけるデータのデータレジスタにおける配置を示す図である。図22において、第2行のBレジスタおよびXレジスタのデータを用いた加算シフト演算操作が実行され、この結果が第3行のBレジスタおよび第2行のA2レジスタへ格納される。次いで第2行の命令実行前に遅延素子の遅延操作が実行される。すなわち、A2レジスタの格納データのA1レジスタへの転送が実行される。具体的には、第1行のA1レジスタの内容が第1行および第4行のA1レジスタへ転送され、第2行のA2レジスタの内容が第1行のA2レジスタ、第2行および第3行のA1レジスタへ転送される。この転送操作実行後上述の加算シフト複合演算操作が実行される。この加算シフト演算操作と並行して、第4行のA1レジスタとCレジスタの乗算が実行され、その乗算結果が第4行のXレジスタに格納される。

【0134】図23は第10実行サイクルにおけるデータレジスタにおけるデータの配置を示す図である。図23において、第3行のA1レジスタとCレジスタの格納データの乗算が行なわれその乗算結果が第3行のXレジスタに格納される。この乗算操作と並行して、第4行のBレジスタとXレジスタの格納データに対する加算およびシフト演算操作が実行され、その複合演算操作結果が出力データとして出力される。

【0135】図24は、第11実行動作サイクルにおけるデータレジスタにおけるデータの配置を示す図である。図24において、第2行のA1レジスタとCレジスタの格納データの乗算が行なわれ、該乗算結果が第2行のXレジスタに格納される。この乗算操作と並行して、第3行のBレジスタおよびXレジスタの格納データに対する加算およびシフト演算操作が実行され、その演算結果が第4行のBレジスタに格納される。

【0136】図25は第12実行サイクルにおけるデータレジスタ内のデータの配置を示す図である。図25において、新たに第1行のDレジスタに次のデータ“3”が入力される。このデータの入力と並行して、第1行のA1レジスタとCレジスタの格納データに対する乗算が行なわれ、その乗算結果が第1行のXレジスタに格納される。またこの入力および乗算操作と並行して、第1行のBレジスタおよびXレジスタに格納されたデータに対する加算およびシフト複合演算操作が実行され、その演算結果が第2行のBレジスタへ格納される。このとき、第1のiir命令における「4(A1)」の操作は、先の図22に示すコピー動作により実行されている。

【0137】図26は第13実行サイクルにおけるデータレジスタにおけるデータの配置を示す図である。第12実行サイクルにおいて第1ないし第4行のA1レジスタのデータはすべて消費されている。このため、A2レジスタからA1レジスタの内容へのデータ転送(コピー動作)が実行される。これにより遅延素子による遅延操

作が実現される。コピー操作後、第4行のデータ（“0”）と第4行のCレジスタとの乗算が行なわれその乗算結果が第4行のXレジスタへ格納される。この乗算操作と並行して、第2行のBレジスタおよびXレジスタの格納データによる加算およびシフト演算操作が実行され、その演算操作結果は第3行のBレジスタおよび第2行のA2レジスタへ格納される。

【0138】図27は、第14実行サイクル完了後のデータレジスタにおけるデータの配置を示す図である。第4行のBレジスタおよびXレジスタの格納データに対し加算およびシフト演算操作が実行され、その演算操作結果が出力データとして出力される。この加算シフト演算操作と並行して、第3行のA1レジスタおよびCレジスタの格納データを利用した乗算が実行され、該乗算結果がXレジスタへ格納される。

【0139】図28(A)は第15実行サイクル完了後のデータレジスタにおけるデータの配置を示す図である。図28(A)において、第2行のA1レジスタとCレジスタの格納データによる乗算が行なわれ、この乗算結果が第2行のXレジスタへ格納される。この乗算操作と並行して、第3行のBレジスタおよびXレジスタの格納データによる加算およびシフト演算操作が実行され、その演算操作結果が第4行のBレジスタへ格納される。この加算およびシフト演算操作と並行して、第2行のA1レジスタとCレジスタの格納データによる乗算が行なわれ、該乗算結果が第2行のXレジスタに格納される。

【0140】図28(B)は、第16実行サイクル完了時におけるデータレジスタ内のデータの配置を示す図である。図28(B)において、まず第1行のBレジスタおよびXレジスタの格納データを用いて加算およびシフト演算操作が実行されその演算操作結果が第2行のBレジスタへ書込まれる。次いで第1行のA1レジスタおよびCレジスタの格納データによる乗算が実行されその乗算結果が第1行のXレジスタ内へ格納される。このXレジスタ内へのデータ格納と並行またはその完了後に新しい入力データ“4”が第1行のBレジスタへ格納される。

【0141】以降、上述の動作が繰り返し実行される。すなわち、第1行のA1レジスタの格納データが2回消費された（読出される）ときに第1行のBレジスタへ新しい入力データが格納される。第1行ないし第4行のA1レジスタのデータがすべて一度消費された場合、A2レジスタからのデータのコピーが実行される。

【0142】上述の動作において、各エントリー（行）において所望のデータが揃ったときに演算が実行される。この演算データが揃ったか否かの判別は、各データレジスタに対する書込制御回路が書込を行なったことを検出することにより行なわれる。必要な2つのデータレジスタへの書込が行なわれたときにデータの読出が行なわれ所望の演算が実行される。複数の行において同一演

算に対して同時にデータが確定した場合には優先順位（行番号の大きい方）に従って演算が実行される。

【0143】このための構成としては以下の構成が考えられる。各レジスタに対応して書込／読出制御回路を設け、この書込／読出制御回路の制御の下に演算器へのデータの伝送および所望データの揃ったことの検出を行なう。どのレジスタへデータを書込むかはEレジスタおよびFレジスタにプログラムされたデータにより決定される。したがってこのプログラムされたデータに従って、書込制御回路が対応のデータレジスタへのデータの書込を実行する。この場合においても、各レジスタに対応して読出制御回路が設けられ、この読出制御回路がEレジスタからの行先情報が自己に割当てられたアドレス（レジスタ位置指定情報）と一致した場合に与えられたデータを書込む。この場合、読出制御回路はデータ読出時において対応のEレジスタまたはFレジスタの内容を読出す構成となる。すなわち、iir命令実行時においては、A1レジスタとCレジスタの内容を読出す場合にはEレジスタの内容も同時に読出され、BレジスタとXレジスタの内容を読出す場合にはFレジスタの内容が同時に読出される。乗算結果の格納は、固定的に定められており、乗算のために選択されたエントリーを指定する情報と同じエントリーのXレジスタが書込可能状態とされデータが書込まれる。

【0144】上述の動作および計算ルール等は、図3に示すシーケンスコントローラSCおよび読出／書込制御部RWCにより実現される。

【0145】図29は、上述の実行サイクル2ないし実行サイクル16における各演算器の稼働状況を示す図である。図29において、○は稼働状態を示し、空白状態はデータ待ち状態すなわち演算器のアイドル状態を示す。

【0146】図29から明らかなように、サイクル5ないし16において、乗算器および加算シフト複合演算器はすべて稼働状態となっており初期動作時を除く安定状態時には演算器が100%動作するフル稼働状態が得られている。すなわち、各動作サイクルにおいて演算器が駆動されており命令実行待ちによるオーバーヘッドは生じず、高速でフィルタ処理を実行することが可能となる。

【0147】上述のフィルタ処理は乗算器1つおよび加算およびシフト複合演算器1つを用いてフィルタ処理を実現している。次に4つの演算器、すなわち4つの乗算器および4つの加算シフト複合演算器を用いてフィルタ処理を行なう場合の動作について説明する。

【0148】図30は4つの演算器によりフィルタ処理する際のデータレジスタにおけるデータの配置を示す図である。図30(A)において、先の図13の場合と同様の状態が初期設定される。図30(A)においては入力データ“1”が第1行のBレジスタに格納された状態

が示される。この状態においては、図 30 (B) に示すようにブロック #1~#4 の加算器の入力がそれぞれ初期設定された状態となっており、また乗算器の入力も初期設定された状態となっている。以下の説明においては、演算器の振舞を観察するために、1 サイクルの動作を加算シフト複合演算器を示す状態 (a) と乗算器を示す状態 (b) の 2 つに分割して観察する。

【0149】図 31 は、第 1 の実行サイクルにおけるデータレジスタにおけるデータの配置およびデジタルフィルタにおけるデータの分布状態を示す図である。図 31 (A) に示すように、第 1 の実行サイクルにおいては、第 1 行ないし第 4 行の A1 レジスタと C レジスタの乗算操作が並列に実行され、その乗算結果が対応の X レジスタへ格納される。シフトおよび加算演算操作は X レジスタに確定データが存在していないため実行されない。

【0150】この状態は図 31 (B) に示すように、各ブロック #1~#4 の乗算器において並列に乗算が実行されその乗算結果が確定した状態を示す。

【0151】図 32 は、第 2 の実行サイクルにおけるデータレジスタ内のデータの配置およびそのときのデジタルフィルタにおけるデータの状態を示す図である。A1 レジスタのデータがすべて消費された状態において、A2 レジスタから A1 レジスタへのコピー動作が実行される。すなわち、図 32 (A) に示すように、まず、第 1 行の A2 レジスタの格納データが第 1 行および第 4 行の A1 レジスタへ格納される。次いで図 32 (B) に示すように第 2 行の A2 レジスタに格納されているデータが第 1 行の A2 レジスタ、第 2 行および第 3 行の A1 レジスタへ転送される。このときまた、第 1 行の B レジスタおよび X レジスタの格納データによる加算およびシフト操作が実行される。

【0152】図 33 は図 32 に示す動作サイクルの次に実行される状態を示す図である。図 33 (A) において、コピー動作完了後、第 1 行ないし第 4 行の B レジスタおよび X レジスタの内容がそれぞれ対応の加算シフト演算器に転送される。この演算結果は、対応の E レジスタに格納された先行情報により示されるレジスタに格納される。第 1 行の B レジスタおよび X レジスタの加算およびシフト結果は第 2 行の B レジスタへ格納される。第 2 行の B レジスタおよび X レジスタの演算結果は第 3 行の B レジスタおよび第 2 行の A2 レジスタへ格納される。第 3 行の B レジスタおよび X レジスタの格納データの加算およびシフト演算結果は第 4 行の B レジスタへ格納される。第 4 行の B レジスタおよび X レジスタの演算結果は出力データとして出力される。この場合、B レジスタに関しては、それぞれの行番号よりも 1 つ大きい行番号の B レジスタへの演算結果の書込が行なわれる。第 2 行に対する演算結果に対してのみ第 2 行の A2 レジスタへもデータが書込まれる。

【0153】このときまた第 1 行の B レジスタへ次の入

力データ (“2”) が書込まれる。この状態により図 33 (B) に示すように各ブロック #1~#4 の加算器の入力が確定状態となりかつ第 2 行の A2 レジスタの内容が確定状態となる。

【0154】図 34 はこの加算およびシフト演算操作完了後のデータレジスタ内のデータの配置を示す図である。図 34 (A) に示すように A1 レジスタの格納データと B レジスタの格納データとの乗算が行なわれ、その乗算結果が X レジスタへ格納される。これにより図 34 (B) に示すように各ブロック #1、#2、#3、および #4 の乗算器の出力がすべて “0” の確定状態となる。ここで、演算は、加算およびシフト演算が乗算よりも先に行なわれる計算ルールが適用されていることを想起されたい。

【0155】図 35 は第 3 実行サイクル状態完了時のデータレジスタ内のデータの配置およびデジタルフィルタ内のデータの状態を示す図である。図 35 (A) は加算およびシフト演算操作実行完了時のデータレジスタ内のデータの配置を示し、図 35 (B) は乗算操作完了後のデータレジスタ内のデータの配置を示す。図 35 (A) には、A2 レジスタから A1 レジスタへの図 32 に示すコピー動作および加算およびシフト演算操作完了後の状態が示される。この状態においてはまた新たに第 1 行の B レジスタへ入力データ “3” が書込まれる。B レジスタと X レジスタのデータを用いて加算およびシフト演算操作が実行され、出力データが出力される。

【0156】図 35 (B) においては、A1 レジスタの格納データと C レジスタの格納データとの乗算が実行されその乗算結果が X レジスタに格納される。

【0157】上述の動作を実行することにより、図 35 (C) に示すように出力データ “0” が得られ、各ブロック #1~#4 の加算器の入力も確定状態となる。続いて、図 36 に示すように、B レジスタ (第 1 行) へデータ “4” が書込まれ、各行の B レジスタの内容が行番号の大きい方へ 1 行シフトされ、かつデータ “0” が出力される。図 36 (A) においても、B レジスタおよび X レジスタのデータによる加算およびシフト演算操作が実行されかつ A2 レジスタから A1 レジスタへのデータのコピー動作も実行されている。コピー動作完了後に加算およびシフト演算操作が実行される。この図 36 (A) の状態完了後図 36 (B) に示す A1 レジスタの内容と C レジスタの内容の乗算が実行される。この乗算結果が各 X レジスタに格納される。以降この動作を繰り返す。

【0158】上述の乗算を並列に実行する構成の場合、各レジスタからは並列にデータが読出されかつ並列にデータが書込まれる。この場合、その各レジスタにおいて先のデータの書込完了後にデータの読出が行なわれ、データの読出は有効データの書込完了後でないと実行されない。

【0159】図 37 はこの 4 つの演算器すなわち 4 つの



25

乗算と4つの加算およびシフト複合演算器を用いた場合の各演算器の稼働状態を示す図である。図37において、乗算はXで示し、加算およびシフト演算操作は「+ shift」で示す。それぞれにおいて、○は演算器が稼働状態にある状態を示し、空白部分は非稼働状態すなわちデータ入力待合わせ状態を示す。図37から明らかなように、実行サイクル2ないし4においては、演算器がすべて稼働状態となっており、高速でフィルタ処理することがわかる。

【0160】[4次FIRフィルタのプログラム実施例] 次に、代表的なフィルタ構成例として4次のFIRフィルタについて考察する。

【0161】図38は4次のFIRデジタルフィルタの構成を示す図である。図38に示す4次FIRデジタルフィルタは、転置型トランスバーサルフィルタとも呼ばれる。図38において、FIRフィルタは、入力データと所定の乗算係数とを乗算する乗算器201、202、203および204と、加算器205、206、207および208と、遅延器209、210および211と除算器212を含む。乗算器201~204は互いに並列に配置される。加算器と遅延器とは交互に配置される。

【0162】加算器205は乗算器201の出力とデータ0とを加算する。遅延器209は加算器205の出力を所定期間（1サンプル周期）遅延する。加算器206は、乗算器202の出力と遅延器209の出力を加算する。遅延器210は、加算器216の出力を所定期間（1サンプル周期）遅延する。加算器207は遅延器210の出力と乗算器203の出力を加算する。遅延器201は加算器207の出力を所定期間遅延する。加算器208は乗算器204の出力と遅延器211の出力とを加算する。除算器212は加算器208の出力を係数4で除算し出力データを生成する。

【0163】図38に示すFIRフィルタでは複数個の同一構成が縦続接続される。このFIRフィルタをfir命令に従ってブロックに分割する。fir命令は、4入力2出力命令であるが、その内部構造として乗算操作、加算操作、シフト操作および遅延操作を含む。したがってこの図38に示すFIRフィルタを乗算器、加算器および遅延素子を含む1つのブロックを単位として分割する。FIRデジタルフィルタはブロック#1~#4に分割され、各ブロック#1~#4がそれぞれfir命令で表現される。

【0164】図39はこのデジタルフィルタをfir命令で記述した場合のフローグラフを示す図である。図39において、第1のfir命令#1は、A入力に定数0を受け、B入力に入力データを受け、C入力に定数0.1を受け、D入力に定数0を受ける。第2のfir命令#2は、A入力に第1のfir命令#1のF出力を受け、B入力に第1のfir命令#1のF出力を受け、B入力に入力データを受け、C入力に定数0.2を

26

受け、D入力に定数0を受ける。第3のfir命令#3は、A入力に第2のfir命令#1のF出力を受け、B入力に第2のfir命令#1のF出力を受け、C入力に定数0.3を受け、D入力に定数0を受ける。第4のfir命令#4は、A入力に第3のfir命令#3のF出力を受け、B入力に第3のfir命令#3のF出力を受け、C入力に定数0.4を受け、D入力に定数2を受ける。第4のfir命令#4のF出力からデータ処理後の出力データが出力される。

【0165】図40は図39に示すフローグラフをテキスト形式で記述したプログラムを示す図である。以下図40を参照してこのプログラムについて説明する。

【0166】図40において左端の番号は行番号を示し、レジスタファイルにおけるエンタリーの行に1対1に対応する。図40のプログラムの右端の番号はプログラム番号を示す。

【0167】プログラム1: プログラム1の「initial」は初期値の宣言を示す。「2(A)=0」は第2行のレジスタAの格納データを「0」に初期設定することを示す。すなわち第2のfir命令#2のA入力、すなわち加算器の入力初期値が「0」であることを示す。同様に「3(A)=4(A)=0」は第3および第4のfir命令#3および#4のA入力、すなわちその加算器の入力値が「0」に初期設定されることを示す。

【0168】プログラム2: 「input」は外部からのデータの入力を示す。「1(B), 2(B), 3(B), 4(B)」は、この入力データが、第1ないし第4のfir命令のB入力へ与えられることを示す。すなわち、レジスタファイルにおいて第1ないし第4行のBレジスタに外部からの入力データが格納されることを示す。

【0169】プログラム3: 「fir」はfir命令を示す。「fir(0, , 0.1, 0)」はこのfir命令における定数を示す。括弧内の値は入力の定数、すなわちfir(A, B, C, D)における入力A、B、CおよびDのそれぞれの定数を表わす。このプログラム番号3のfir命令においては、Aの定数値が0、Bへは定数値でなく変数が与えられ、C入力の定数値が0.5に設定され、Dの定数値が0に設定される。レジスタファイルにおいては、定数は変更されない。複数回読出が可能である。B入力には入力データが与えられ、このデータを格納するレジスタBにおいては単一代入単一読出のルールが適用される。

【0170】「absorb! 2(A)」はE出力およびF出力の行先を示す。この場合、「absorb」はE出力はどの命令へも伝達されないことを示し、F出力が第2のfir命令のA入力へ与えられることを示す。すなわちレジスタファイルにおいて第2行のAレジスタへ格納されることを示す。E出力とF出力とは「!」で区別される。

【0171】プログラム4、5: 命令firの定数の定



義は上で示したプログラム 3 のそれと同じであり、その行先についてもプログラム 3 の場合と同じである。すなわち、第 2 の f i r 命令の F 出力は第 3 の f i r 命令の A 入力へ与えられ、第 3 の f i r 命令の F 出力は第 4 の f i r 命令の A 入力へ与えられる。

【0172】プログラム 6 : f i r 命令の定数については上と同じである。出力における「output」はこの第 4 の f i r 命令の F 出力が装置外部へ出力され、出力データとして利用されることを示す。

【0173】[FIR フィルタ操作の具体的動作] F I R フィルタ処理の場合においても以下のルールが適用される。

【0174】[基本ルール] 第 1 行のエントリーの A レジスタには定数が格納される。第 2 行ないし第 4 行の A レジスタの値は初期値であり、新たに与えられるデータにより更新される。また第 1 ないし第 4 行の C レジスタおよび D レジスタの格納値は定数であるため、これらのデータは更新されない。

【0175】定数以外のデータには、すべてのレジスタにおいて 1 回書込 / 1 回読出すなわち単一代入単一読出のルールが適用される。

【0176】E レジスタに格納される行先情報および F レジスタに格納される行先情報により、複数かつ任意の行先レジスタを指定することができる（プログラム可能である）。この F I R フィルタ処理においては B レジスタの格納データは E レジスタの行先情報により指定された行先へ伝達されるが、この場合「absorb」であるため出力されない。シフト操作結果は、F レジスタに格納された行先情報が特定するレジスタへ伝達される。

【0177】乗算を行なう演算器が 1 つ、および加算およびシフトを含む演算器が 1 つ存在する。

【0178】[計算ルール] i i r 命令の場合と同様である。すなわち、

(1) 演算は行番号にかかわらず必要データが揃ったものから順次実行される。

【0179】(2) 演算は加算およびシフトが乗算よりも優先的に実行される。

(3) 同一の演算に対しデータが同時に揃った場合には、行番号の逆順、すなわち大きい行番号の命令に対する操作が実行される。

【0180】(4) 加算およびシフト複合演算において F レジスタが複数の行先を指定する場合、行先のレジスタがすべて空状態でない限りその演算は実行されない。

【0181】(5) 後からの演算が前の演算より優先されることはない（演算の追越しは生じない）。

【0182】次に具体的動作について説明する。

(0) 初期設定

図 4 1 はレジスタファイルにおける各レジスタの初期状態と演算器との接続関係およびそのときの F I R デジタ

ルフィルタにおけるデータの配置を示す図である。

【0183】図 4 1 (A) において、初期状態においては、第 1 行ないし第 4 行の C レジスタには乗算係数が定数データとして格納され、また第 1 行ないし第 4 行の D レジスタにはシフト操作におけるシフトビット数を示すためのデータが定数データとして格納される。

【0184】F I R フィルタ処理においては、B レジスタの格納データと C レジスタの格納データとが乗算器へ与えられ、乗算操作を受ける。A レジスタと X レジスタのデータが加算操作を受け、次いでシフト操作を受ける。乗算操作の結果は X レジスタへ格納される。加算およびシフト操作の演算結果は F レジスタに格納された行先情報が特定するレジスタへ書込まれる。

【0185】図 4 1 (B) に示すように、初期状態においては、乗算器における乗算係数（フィルタ係数）が設定されかつブロック # 4 における除算器の除算係数が初期設定される。係数 4 で除算をするため、シフト操作により 2 ビット下位ビット方向へデータがシフトされる。

【0186】A レジスタに格納されたデータはそれぞれ加算器の一方入力のデータを与える。ブロック # 1 における加算器の一方入力は定数“0”に固定される。

【0187】(1) 実行サイクル 1

図 4 2 は実行サイクル 1 における各レジスタのデータの配置およびそのときのデジタルフィルタにおけるデータの配置を示す図である。この実行サイクル 1（実行状態 1）においては、外部から入力データ“1”が与えられ、第 1 行ないし第 4 行の B レジスタにこの入力データが格納される。この状態においては、図 4 2 (B) に示すように、乗算器の入力が確定した状態となる。

【0188】(2) 実行サイクル 2

図 4 3 は実行サイクル 2（実行状態 2）のレジスタファイルにおける各レジスタのデータの配置およびそのときのレジスタフィルタのデータの分布を示す図である。

【0189】この実行サイクル 2 においてはまず第 4 行の B レジスタの格納データと C レジスタの格納データの乗算が実行される。乗算に必要なデータが揃いかつ一番行番号の大きい行に対する乗算が実行される計算ルールに従っている。この乗算結果は第 4 行の X レジスタに格納される。すなわち、図 4 3 (B) に示すように、ブロック # 4 において乗算器の出力が確定状態となる。

【0190】(3) 実行サイクル 3

図 4 4 は実行サイクル 3（実行状態 3）におけるレジスタのデータの配置およびそのときのフィルタにおけるデータの分布を示す図である。実行サイクル 3 においては、まず第 4 行のエントリーにおいて、A レジスタおよび X レジスタのデータが揃ったためこれらのデータを用いて加算およびシフト演算操作が実行される。この加算およびシフト演算操作により出力データが生成される。この加算およびシフト演算操作と平行して、第 3 行の B レジスタの格納データと C レジスタの格納データとの乗

算が実行され、その乗算結果が第3行のXレジスタに格納される。すなわち、図44(B)に示すように、ブロック#4において、加算および除算が実行され、出力データ“0.1”が生成される。これと並行して、ブロック#3において、乗算が実行され、その乗算結果“0.3”がXレジスタに格納される。

#### 【0191】(4) 実行サイクル4

図45は実行サイクル4(実行状態4)におけるレジスタファイルのデータの格納状態およびデジタルフィルタのデータの分布を示す図である。図45(A)に示すように、第2行のBレジスタの格納データとCレジスタの格納データとの乗算が実行され、その乗算結果が第2行のXレジスタに格納される。この乗算操作と並行して、第3行のAレジスタおよびXレジスタのデータを用いて加算およびシフト演算操作が実行され、その演算結果が第4行のAレジスタに格納される。

【0192】すなわち、図45(B)に示すように、ブロック#3における加算および遅延が実行される。

【0193】iir命令の場合、遅延操作はレジスタ間のデータ転送により実現される。このfir命令においては、遅延操作は、前段のブロックの加算およびシフト操作が1サイクル遅れて実行されているため、等価的に表現されていることになる。

【0194】同様に、ブロック#2において、乗算が実行されており、その乗算器の出力が確定状態となる。

#### 【0195】(5) 実行サイクル5

図46は実行サイクル5(実行状態5)の完了後のレジスタファイルにおける各レジスタの格納データおよびそのときのデジタルフィルタのデータの分布を示す図である。

【0196】図46(A)に示すように、まず第1行のBレジスタとAレジスタ、Cレジスタの格納データを用いて乗算が実行され、その乗算結果が第1行のXレジスタに格納される。この乗算と並行して、第2行のAレジスタとXレジスタの格納データを用いた加算およびシフト演算操作が実行され、その演算操作結果が第3行のAレジスタに格納される。

【0197】この乗算および加算・シフト演算操作と並行して、第1行ないし第4行のBレジスタに新たな入力データ“2”が格納される。

【0198】すなわち図46(B)に示すように、ブロック#1において乗算器の出力が確定し、またブロック#2において、その遅延素子の出力が確定状態となり、かつさらにブロック#1～#4の乗算器の入力が確定状態となる。

#### 【0199】(6) 実行サイクル6

図47は実行サイクル6(実行状態6)における各レジスタ格納データおよびそのときのデジタルフィルタにおけるデータの分布を示す図である。図47(A)に示すように、第4行のBレジスタとCレジスタの格納データ

を用いて乗算が実行され、その乗算結果が第4行のXレジスタに格納される。この乗算操作と並行して、第1行のAレジスタとXレジスタの格納データを用いて加算およびシフト演算操作が実行され、その演算結果が第2行のAレジスタに格納される。

【0200】すなわち、図47(B)に示すように、ブロック#1の遅延素子の出力が新たに確定状態となり、またブロック#4において、乗算器の出力が確定状態となる。

#### 【0201】(7) 実行サイクル7

図48は実行サイクル7における各レジスタの格納データおよびそのときのデジタルフィルタにおけるデータの分布を示す図である。図48(A)に示すように、第4行のAレジスタとXレジスタの格納データを用いて加算およびシフト演算が実行され、その演算結果が出力データとして出力される。

【0202】この加算およびシフト演算操作と並行して、第3行のBレジスタとCレジスタの格納データを用いて乗算が実行され、その乗算結果が第3行のXレジスタに格納される。

【0203】すなわち、図48(B)に示すように、ブロック#3において乗算器の出力が確定し、またブロック#4において加算および除算が実行され出力データが生成される。

#### 【0204】(8) 実行サイクル8

図49は実行サイクル8(実行状態8)のレジスタファイルの各レジスタの格納データおよびそのときのデジタルフィルタにおけるデータの分布を示す図である。

【0205】図49(A)に示すように、第3行のAレジスタおよびXレジスタの格納データを用いて加算およびシフト演算操作が実行され、その演算結果が第4行のAレジスタに格納される。この加算およびシフト演算操作と並行して、第2行のBレジスタおよびCレジスタの格納データを用いて乗算が実行され、その乗算結果が第2行のXレジスタに格納される。

【0206】すなわち図49(B)に示すように、ブロック#2において乗算器の出力データが確定し、またブロック#3において、遅延素子の出力データが確定する。

#### 【0207】(9) 実行サイクル9

図50は実行サイクル9(実行状態9)の各レジスタの格納データおよびそのときのデジタルフィルタにおけるデータの分布を示す図である。図50(A)に示すように、第1行のBレジスタおよびCレジスタの格納データを用いて乗算が実行され、その乗算結果が第1行のXレジスタに格納される。また、第2行のAレジスタとXレジスタの格納データを用いて加算およびシフト演算が実行され、その演算結果が第3行のAレジスタに格納される。これらの演算と並行して、第1行ないし第4行のBレジスタに次の入力データが格納される。

【0208】すなわち、図50(B)に示すように、ブロック#1において乗算器のデータが確定し、ブロック#2の遅延素子の出力データが確定し、またブロック#1~#4の乗算器の入力が確定状態となる。

#### 【0209】(10) 実行サイクル10

図51は実行サイクル10(実行状態10)のレジスタの格納データおよびそのときのデジタルフィルタのデータの分布を示す図である。図51(A)に示すように、まず第4行のBレジスタおよびCレジスタの格納データを用いた乗算が実行され、その乗算結果が第4行のXレ

ジスタに格納される。

【0210】また第1行のAレジスタおよびXレジスタの格納データを用いて加算およびシフト演算が実行され、その演算結果が第2行のAレジスタに格納される。

【0211】すなわち図51(B)に示すように、ブロック#1の遅延素子の出力データが確定状態となり、またブロック#4の乗算器の出力データが確定状態となる。

#### 【0212】(11) 実行サイクル11

図52は実行サイクル11(実行状態11)の各レジスタの格納データおよびそのときのデジタルフィルタにおけるデータの分布を示す図である。図52(A)に示すように、まず第4行のAレジスタおよびXレジスタの格納データを用いて加算およびシフト演算が実行され、その演算結果が出力データとして出力される。この加算およびシフト演算と並行して、第3行のBレジスタとCレジスタの格納データを用いた乗算が実行され、その乗算結果が第3行のXレジスタに格納される。

【0213】すなわち、ブロック#3における乗算器の出力データが確定状態となり、またブロック#4から出力データが出力される。

#### 【0214】(12) 実行サイクル12

図53は実行サイクル12(実行状態12)におけるレジスタの格納データおよびデジタルフィルタにおけるデータの分布を示す図である。図53(A)に示すように、まず第2行のBレジスタおよびCレジスタの格納データを用いて乗算が実行され、その乗算結果が第2行のXレジスタに格納される。この乗算操作と並行して、第3行のAレジスタおよびXレジスタを用いて加算およびシフト操作が実行され、その演算結果が第4行のAレ

ジスタに格納される。

【0215】すなわち図53(B)に示すように、ブロック#2における乗算器の出力が確定し、またブロック#3の遅延素子の出力が確定状態となる。

#### 【0216】(13) 実行サイクル13

図54は実行サイクル13(実行状態13)における各レジスタの格納データおよびそのときのデジタルフィルタにおけるデータの分布を示す図である。図54(A)において、第1行のBレジスタおよびCレジスタの格納データを用いて乗算が実行され、その乗算結果が第1行

のXレジスタに格納される。この乗算と並行して、第2行のAレジスタおよびXレジスタの格納データを用いて加算およびシフト演算が実行され、その演算結果が第3行のAレジスタに格納される。これらの演算と並行して、第1行ないし第4行のB入力に新たな入力データが格納される。

【0217】すなわち図54(B)に示すようにブロック#1~#4の乗算器の入力が確定状態となり、またブロック#1の乗算器の出力が確定し、またブロック#2の遅延素子の出力が確定状態となる。

#### 【0218】(14) 実行サイクル14

図55は、実行サイクル14(実行状態14)におけるレジスタの格納データおよびそのときのデジタルフィルタのデータの分布を示す図である。図55(A)において、第4行のBレジスタおよびCレジスタの格納データを用いて乗算が実行され、その乗算結果が第4行のXレジスタに格納される。この乗算と並行して、第1行のAレジスタおよびXレジスタの格納データを用いて加算およびシフト演算が実行され、その演算結果が第2行のAレジスタに格納される。

【0219】すなわち図55(B)に示すように、ブロック#1の遅延素子の出力が確定状態となり、またブロック#4の乗算器の出力が確定状態となる。

#### 【0220】(15) 実行サイクル15

図56は実行サイクル15(実行状態15)におけるレジスタの格納データおよびそのときのデジタルフィルタにおけるデータの分布を示す図である。図56(A)において、まず第4行のAレジスタおよびXレジスタの格納データを用いて加算およびシフト演算が実行され、その演算結果は出力データとして出力される。この加算およびシフト演算と並行して、第3行のBレジスタとCレジスタの格納データを用いて乗算が実行され、その乗算結果が第3行のXレジスタに格納される。

【0221】すなわち図56(B)に示すように、ブロック#4から出力データが生成されて出力され、かつブロック#3において乗算器の出力が確定状態となる。

【0222】上述のように、1つの乗算器と1つの加算およびシフト演算器を用いる構成においては、行番号の大きいエンタリーのデータを用いてまず乗算が実行され、次いで加算およびシフト演算が実行される。この操作を、加算およびシフト演算操作を乗算操作と並行に実行し、Bレジスタに格納されるデータがすべて消費されるまで順次実行する。これにより、FIRフィルタ処理を高速で実行することができる。

【0223】図57は前述のサイクル2ないし16における各演算器の稼働状況を示す図である。図57において○印は稼働状態を示し、空白はデータ待ち状態を示す。図57に示されるように、第4サイクルないし第16サイクルにおいて乗算器(X)および加算およびシフト複合演算器(+shift)はすべて稼働状態にあ

り、初期動作時を除く安定時においては演算器は100%のフル稼働状態となっており、命令実行のオーバーヘッドなく高速でフィルタ処理が行なわれているのが見られる。

【0224】上述の説明においては、1個の乗算器および1個の加算シフト演算器が利用されている。この場合4個の乗算器および4個の加算シフト演算器を用いてフィルタ処理を実行することができる。以下この場合の構成について説明する。

#### 【0225】(0) 初期状態

図58は、4個の乗算器と、4個の加算およびシフト演算器を用いた場合の各レジスタの初期状態を示す図である。初期状態においては、第1行ないし第4行のBレジスタに入力データが格納される。またこのとき、CレジスタおよびDレジスタにおいて乗算係数およびシフト係数が定数データとして初期設定される。さらに、Aレジスタにおいては、その初期値が設定される。

#### 【0226】(1) 実行サイクル1

図59は実行サイクル1におけるレジスタの各格納データとそのときのデジタルフィルタにおけるデータの分布を示す図である。実行サイクル1(実行状態1)においては、加算およびシフト演算動作は実行されない。乗算動作のみが実行される。すなわち図59(A)に示すように、第1行ないし第4行のBレジスタとCレジスタの格納データの乗算が行なわれ、各乗算結果が対応のXレジスタに格納される。

【0227】図59(B)に示すように、ブロック#1~#4各々において乗算が並列に実行され、各乗算器の出力データが確定状態となる。

#### 【0228】(2) 実行サイクル2

図60は実行サイクル2におけるレジスタの格納データおよびそのときのデジタルフィルタにおけるデータの分布を示す図である。図60(A)において、第1ないし第4行のAレジスタおよびXレジスタの格納データを用いて加算およびシフト演算が実行される。このとき、演算結果は次のエントリ(次の行)のAレジスタに格納される。第4行のAレジスタおよびCレジスタのデータの演算結果は出力データとして出力される。

【0229】すなわち図60(B)に示すように、各ブロック#1~#4各々において、遅延素子の出力データが確定状態となるとともに、ブロック#4においては、除算器の出力が確定状態となり、出力データが生成される。

【0230】この第2サイクルにおいてはBレジスタには有効データは存在しないため、乗算は実行されない。すなわち、複数個の乗算器と複数個の加算およびシフト演算器を利用する場合には、乗算が行なわれ、次いで加算およびシフト演算が実行される。この動作が繰り返し新しい入力データに対して実行される。

【0231】この第2サイクルにおいて、加算およびシ

フト演算と並行して、Bレジスタへのデータの格納および乗算が並行に実行される。すなわち、第1サイクルにおいて、乗算が行なわれると次いで第1行ないし第4行のBレジスタにデータが入力される。したがって、この場合、第2サイクル以降において、乗算器と加算シフト演算器は常時稼働状態となり、演算器が入力データ待合わせ状態となることはなく、高速でフィルタ処理を実行することが可能となる。

【0232】ここで図59(B)および図60(B)において、括弧で示したデータは、それぞれのサイクルにおいて並行して実行された操作の結果得られるデータを示す。

【0233】レジスタとバス(演算器とレジスタとを結合するバス)との接続はプログラムにより決定される。複数の演算器、すなわち複数の乗算器と複数の加算およびシフト演算器を用いる構成の場合、各行並列に動作するため、各行のレジスタと演算器との接続はプログラムにより一意的に固定される。この場合、同一種類の演算において各行の優先順位を特に判別する必要はない。

【0234】しかしながら、1個の乗算器と1個の加算およびシフト複合演算器を利用する構成の場合、1つの演算(乗算または加算シフト演算)を行なう場合、各行において優先順位を判別する必要がある。この優先順位を判別し、必要なデータが揃った場合にデータを読出す構成について以下に説明する。

【0235】図61は、レジスタファイルにおける各レジスタの書込/読出制御を行なう構成を示す図である。この図61は、図3に示す書込/読出制御部の構成をより具体的に示す。

【0236】図61において、代表的に2種類のレジスタを示す。この2種類のレジスタは同一の演算に使用されるデータを格納する。

【0237】レジスタ#11に対し、書込ゲート311aおよび読出ゲート312aが設けられる。レジスタ#12に対し書込ゲート311bおよび読出ゲート312bが設けられる。

【0238】第2行において、レジスタ#21に対し書込ゲート311cおよび読出ゲート312cが設けられる。同様にレジスタ#12に対し、書込ゲート311dおよび読出ゲート312dが設けられる。

【0239】第3行において、レジスタ#31に対し書込ゲート311eおよび読出ゲート312eが設けられる。レジスタ#32に対し書込ゲート311fおよび読出ゲート312fが設けられる。

【0240】第4行において、レジスタ#41に対し書込ゲート311gおよび読出ゲート312gが設けられ、レジスタ#42に対し書込ゲート311hおよび読出ゲート312hが設けられる。

【0241】書込ゲート311a~311hはそれぞれライトイネーブル信号WE11、WE12、WE21、

WE22、WE31、WE32、WE41およびWE42にตอบสนองしてそれぞれ活性状態となり、関連の書込データバス362aまたは362b上のデータを対応のレジスタへ書込む。

【0242】各行において、レジスタの読出を制御するために読出制御回路315a～315dが設けられる。

【0243】第4行の読出制御回路315dはライトイネーブル信号WE41およびWE42がともに活性状態となり、レジスタ#41および#42にデータが書込まれたときに活性状態となり、対応の読出ゲート312gおよび312hをイネーブルし、レジスタ#41および#42をそれぞれ読出データバス360aおよび360bへ接続する。読出データバス360aおよび360bは演算器350に結合される。演算器350は乗算器または加算シフト複合演算器である。

【0244】第3行の読出制御回路315cは、ライトイネーブル信号WE31およびWE32がともに活性状態となり、かつ読出制御回路315dがデータ読出を指示していないときに活性化され、読出ゲート312eおよび312fを活性状態とする。

【0245】第2行の読出制御回路315bは、ライトイネーブル信号WE21、WE22が活性状態となりかつ、読出制御回路315cおよび315dがともにデータ読出を指示していないときに活性化され、対応の読出ゲート312cおよび312dを活性状態とする。

【0246】第1行の読出制御回路315aは、ライトイネーブル信号WE11およびWE12の活性状態になり、かつ、読出制御回路315b、315cおよび315dがデータ読出を指示していないときに活性状態となり、対応の読出ゲート312aおよび312bを活性状態とする。

【0247】演算器350の出力はレジスタ選択回路350を介して所望のデータバスへ伝達される。いずれのデータバスへこの演算器350の出力が伝達されるかはプログラムにより決定される。レジスタ選択回路351は、EレジスタまたはFレジスタの内容をデコードし、対応の書込ゲートを活性化するためのライトイネーブル信号WEを発生するとともに、演算器350からのデータを対応の書込データバス上に伝達する。演算器350の出力がいずれの書込データバス上に伝達されるかは実現されるフィルタの構造により決定される。演算器350が乗算器の場合には、この乗算器出力はXレジスタへ出力されるため、Xレジスタ対応の書込データバスへ伝達される。この場合、レジスタ選択回路351は、別の構成が用いられ、EレジスタまたはFレジスタにかかわりなく、読出制御回路315a～315dの出力に応じて、データ読出を受けた行に対応するXレジスタへデータを伝達する。

【0248】上述の構成において、読出制御回路315a～315dは下位の読出制御回路（行番号の大きい読

出制御回路）がデータ読出を行っていないときにのみライトイネーブル信号にตอบสนองして読出ゲートをイネーブル状態とする。それにより、常に優先順位に従ってデータの読出を実行することができるとともに、演算に必要なデータが揃ったときには即座に演算を実行することができる。

【0249】図62は読出制御回路の具体的構成の一例を示す図である。図62(A)において、読出制御回路315は、一方のライトイネーブル信号WEa1をそのS入力に受けるセット・リセット(SR)フリップフロップ401と、他方のライトイネーブル信号WEa2をそのS入力に受けるSRフリップフロップ402と、SRフリップフロップ401および402の出力と、下位側の読出制御回路のインバータ回路の出力を受けるAND回路403と、AND回路403の出力を反転して上位の読出制御回路に含まれるAND回路の入力へ与えるインバータ回路405と、AND回路403の出力をS入力に受けるSRフリップフロップ404と、SRフリップフロップ404のQ出力を所定時間遅延させる遅延回路406を含む。

【0250】遅延回路406の出力はフリップフロップ401、402および404のリセット入力Rへ与えられる。またフリップフロップ404のQ出力は対応の読出ゲートへ読出イネーブル信号として与えられる。次に動作についてその動作波形図である図62(B)を参照して説明する。

【0251】下位のインバータ回路の出力が“L”の場合には、下位のレジスタ（行番号の大きいレジスタ）においてデータが揃っており先に下位側において演算が実行されることを示す。したがってこの場合、ライトイネーブル信号WEa1およびWEa2がともに活性状態となり対応のレジスタに有効データが書込まれ、フリップフロップ401および402がセット状態となった場合でもAND回路403の出力は“L”状態にある。

【0252】下位のインバータ回路の出力が“H”に立上がると、この行のレジスタの格納データを用いて演算が実行されるべきことを示す。このとき、AND回路403の出力が“H”に立上がり、インバータ回路405の出力が“L”に立下がる。インバータ回路405の出力が上位のAND回路の入力に与えられている。これにより上位側における演算が禁止される。

【0253】AND回路403の出力が“H”に立上がると、フリップフロップ404のQ出力が“H”へ立上がり、対応の読出ゲートがイネーブル状態となり、レジスタの内容が読出データバスに伝達される。所定時間が経過すると、遅延回路406の出力よりフリップフロップ401、402および404がリセットされ、読出ゲートはディスエーブル状態とされるとともに、AND回路403の出力も“L”に立下がり、上位の読出制御回路を読出可能状態に設定する（インバータ回路405の

出力が“H”となる)。この図62に示す構成において、フリップフロップ404はAND回路403の“H”の信号をセット入力Sに受けているときにリセット入力Rに“H”の信号を受けてリセット状態とされる。この構成は、セット状態にする駆動トランジスタよりもリセット状態に設定するトランジスタの方の駆動能力を大きく設定することにより実現される。すなわち、セット能力よりもリセット能力の方が大きくされる。

【0254】この構成において、一方のレジスタが定数データを格納する場合には、フリップフロップ401または402のQ出力を常時“H”状態に設定する必要がある。この構成は、定数データを格納するレジスタに対応するフリップフロップのリセット入力Rに対しては、遅延回路406の出力が伝達されないようにプログラムにより固定する。すなわち、遅延回路406とフリップフロップ401または402との間にスイッチ回路を設けておき、このスイッチ回路の導通/遮断を定数値であるか否かを示すプログラムにより決定する。

【0255】上述の構成により、優先順位を乱すことなく必要なデータが揃ったときに演算を実行することが可能となる。ここで、乗算操作と加算シフト複合演算操作とでは、加算およびシフト演算操作が優先される。この場合、加算シフト演算操作に利用されるレジスタに対する読出制御回路の出力を、乗算回路に利用されるデータを格納するレジスタに設けられる読出制御回路の活性/不活性を指定する信号として利用すればよい（すなわちAND回路403の入力へ与えればよい）。

【0256】〔システム構成例〕これまで述べてきたフィルタ装置においては、レジスタファイル内において必要なデータが揃ったときに演算処理が実行される。このようなデータ処理方式は「データ駆動方式」と呼ばれる。

【0257】一方において、データ処理システムにおける処理能力を増大させるための一手法として、並列処理能力を有するデータ駆動型処理装置が知られている。データ駆動型処理装置においては、「ある処理に必要なデータがすべて揃い、かつその処理に必要な演算装置などの資源が割当てられたときにその処理を実行する」という規則に従って処理が進行する。

【0258】データ駆動型処理装置およびデータ駆動型フィルタ装置はともに「データ駆動方式」に従って処理を実行する。両装置を組合せて1つの情報処理装置を構築した場合、すべての処理を「データ駆動方式」に従って実行するデータ駆動型情報処理装置が実現される。

【0259】このようなデータ駆動型情報処理装置を用いれば、たとえば高速フーリエ変換(FFT)処理を含む画像データ処理などのようなフィルタ処理を含む複雑なプログラムであっても高速かつ効率的に実行することが可能となる。高速フーリエ変換(FFT)は基本構成要素がバタフライ演算要素であり、このバタフライ演算

要素は、遅延素子と加算素子とを基本構成要素として含む。したがってこのようなバタフライ演算要素はfir命令で記述することができるため、高速フーリエ変換処理を、fir命令の組合せにより実行することができるからである。以下に、このデータ駆動型フィルタ装置とデータ駆動型処理装置を組合せる構成について説明する。

【0260】図1はこの発明の一実施例によるデータ駆動型情報処理装置の全体の構成を概略的に示す図である。図1において、データ駆動型情報処理装置500は、データ駆動型フィルタ装置部分510と、データ駆動型プロセッサ部分550とを含む。

【0261】データ駆動型フィルタ装置部分510は、複数の乗算器1a~1nを含む第1の演算器群40と、複数の加算およびシフト演算器2a~2mを含む第2の演算器群50と、第1および第2の演算器群40および50とのデータの授受を行なうレジスタファイル10を含む。図1に示すデータ駆動型フィルタ装置部分510は、図2に示すフィルタ装置と実質的に同様の構成を備える。レジスタファイル10は図3に示すものと同様の構成を備える（この構成については後に再度説明する）。

【0262】データ駆動型プロセッサ部分550は、外部から与えられるデータパケットを受け内部データパケットを生成する入力部556と、実行されるべきプログラムの命令をプログラムのノードに対応して格納するプログラムメモリおよびこの入力データが発火状態にあるか否かに関する情報を記憶する待ち合わせメモリを含むプログラム/待ち合わせメモリ部552と、プログラム/待ち合わせメモリ部552からの発火演算データに対し演算を実行する演算部554と、プログラム/待ち合わせメモリ部552からのメモリアクセス要求命令に従って、データメモリ570へアクセスするメモリインターフェース(メモリI/F)560と、プログラム/待ち合わせメモリ部552から出力されるデータパケットを受け装置外部へ出力するための外部データパケットを生成する出力部558を含む。

【0263】データ駆動型プロセッサにおいては、一般にデータはパケットの形態で伝播される。このパケットは命令と、入力データが到達すべき行先(プログラムノード)を示すノード番号と、発火検出のための入力データの組みを識別するための世代番号などの情報を含む。ある処理に必要なとされるデータがすべてそろった状態を「発火」と称する。この発火検出の際に、ある処理に対しては1組の入力データしか許さないものは静的データ駆動方式と呼び、一方、2組以上の入力データの組みを許すものが動的データ駆動方式と呼ばれる。プログラム/待ち合わせメモリ部552は、命令コードと次の命令フェッチに必要な行先情報(ノード特定番号)両者の付替を行なう機能と、ある処理に必要なデータが揃っ



たこと（発火）の検出を行なう機能とを備える。

【0264】この命令コードおよびノード番号の付替のための情報はプログラムメモリ部に格納され、発火検出に必要な情報は待ち合わせメモリに格納される。このプログラム／待ち合わせメモリ部552において発火が検出されたデータはその行先に従って演算部554、メモリインターフェース560またはフィルタ装置510内のレジスタファイル10へ伝達される。処理が必要とされないデータは出力部558を介して装置外部へ出力される。次に簡単に動作について説明する。

【0265】入力部556からの入力データパケットは、プログラム／待ち合わせメモリ部552へ与えられる。このプログラム／待ち合わせメモリ部552において、プログラムメモリから対応の命令が読出され、この命令の実行の可能／不可能が待ち合わせメモリに格納されたデータにより判別される。実行可能となったとき

（発火状態となったとき）、その命令がフィルタ操作実行命令の場合にはフィルタ装置510のレジスタファイル10へ伝達され、それ以外の命令の場合には、演算部554へ伝達され数値演算／論理演算が実行されるかまたはメモリインターフェース部560へ与えられてデータメモリ570に対するデータの書込／読出が実行されるかまたはさらに出力部558を介してこのプロセッサ550の外部へ送出される。

【0266】データ駆動型フィルタ装置510内で処理されるデータは入力データが与えられた場合、このフィルタ装置510内でフィルタ処理され、結果データのみが再びプログラム／待ち合わせメモリ部552へ伝達される。一方プロセッサ550内で処理されるデータは、プログラム／待ち合わせメモリ部552、演算部554、メモリインターフェース560の間を伝達されることにより発火の有無に従って処理が進行する。フィルタ装置510で実行されるべき命令はこのプログラム／待ち合わせメモリ部552において無条件で発火状態とされ、発火パケットが生成されレジスタファイル10へ伝達される。次に各部の構成の詳細について順次説明する。

【0267】図63は図1に示すプログラム／待ち合わせメモリ部の概略構成を示す図である。図63において、プログラム／待ち合わせメモリ部552は、図1に示す入力部556、メモリインターフェース（I/F）560、フィルタ装置550、および演算部554からのデータパケットの合流を行なうための合流器580と、データの流れの揺らぎを吸収するためのキューバッファ582と、キューバッファ582からのデータに対し、命令コードおよびノード特定番号の付替および発火検出を行なうためのプログラム機能付発火制御部（FCP部）584と、FCP部584からのデータをフィルタ装置510または分岐器588へ分岐させるための分岐器586と、分岐器586からのデータパケットを演

算部554、出力部558およびメモリインターフェース560へ分岐させるための分岐器588を含む。合流器580は、4入力1出力の合流器であり、その出力に空きが生じていることを条件としてその入力に与えられたデータを出力する。

【0268】分岐器586および588は、データパケットに含まれる行先ノード特定番号に従って与えられたデータの振り分けを実行する。

【0269】図64は図63に示す合流器へ与えられるデータパケットのフォーマットを示す図である。図64において、データパケットは、ワードの先頭位置を示すワード識別フラグH/Tと、外部制御系へ転送されるべきデータであることを示す外部制御系転送フラグCTLをその先頭ビット位置に含む。データパケットはさらに、命令コードOPCと、この命令の内容を示すフラグFLAGを含む。このフラグFLAGは、命令がメモリアクセス命令であるか否か、フィルタ操作命令であるか否か、およびこのデータ駆動型プロセッサ内の演算部554で実行される命令であるかなどを示すフラグを含む。すなわちこのフラグFLAGは複数ビットである。フラグFLAGはさらに、命令コードOPCが示す命令が2項演算命令であるか単項演算命令であることを示すフラグをも含む。

【0270】データパケットはさらに、このデータパケットの行先を示す行先データを格納する行先フィールドと、このデータの組を識別するための世代番号を格納する世代フィールドと、データが格納されるオペランドフィールドとを含む。行先フィールドは、このデータパケットがノードの右側ノードへ与えられるか左側ノードへ与えられるかの2入力ノードの左右識別を示すフラグL/Rと、このプロセッサ内で実行されるプログラムのノード特定番号を示すノード番号ND#およびマルチプロセッサシステムにおけるプロセッサを特定するためのプロセッサ番号PE#を含む。世代フィールドは世代番号GE#を含む。この世代番号GE#は、プロセッサとして動的データ駆動型プロセッサを想定しているために用いられる。待ち合わせ時において行先が同一でありかつ世代番号が同じデータが揃ったときに発火と判定される。

【0271】オペランドフィールドはオペランドデータDATAを含む。このオペランドデータは被処理データである。

【0272】図63に示す構成においては、まず合流器580においてデータパケットが合流されてキューバッファ582へ与えられた後、順次このデータパケットがFCP部584へ与えられる。FCP部584においては、行先フィールドのノード番号ND#と世代番号GE#とから構成されるハッシュアドレスをプログラムメモリおよび待ち合わせメモリのアドレスとして対応の命令およびその命令実行後に到達すべきノード番号とが読出



され、次に実行されるべき命令が確定される。

【0273】この次に実行される命令が待ち合わせメモリ部において待ち合わせが完了したと判断され実行可能とされた場合には、この命令は分岐器586によりフィルタ装置510へ与えられるか分岐器588を介して演算部554またはメモリインターフェース560または出力部558へ伝達される。発火条件については後に詳細に説明するが、フィルタ装置510へ与えられて実行されるべき命令の場合には無条件で発火状態とされ、そのノード番号とともに新たな命令コード（フィルタ操作命令）がデータパケットに付され、フィルタ装置510へ与えられる。このフィルタ装置で実行される命令であるか演算部554で実行される命令であるかまたはメモリインターフェースもしくは出力部へ伝達される命令であるかは命令フィールドに含まれるフラグFLAGを見ることにより判別される。

【0274】フィルタ装置ではこの命令フィールドに含まれる命令コードOPCをデコードし（フィルタ操作は関数呼出の形で実現される）、この命令コードが指定するフィルタ演算に対応するプログラムを図3に示すプログラムメモリPMからシーケンスコントローラSCが読出してレジスタファイルにセットし、この命令に従ったフィルタ操作が順次実行される。

【0275】図65は図63に示すFCP部のより詳細な構成を示す図である。図65において、FCP部584は、たとえば連想メモリを用いて構成されるFCPメモリ600と、6段のマイクロバイラインを構成するように配置された制御ロジックを含む。FCPメモリ600は、入力データパケットに含まれる行先ノードに対する命令コードと、そのノードの演算結果が伝達される行先ノード番号を格納するPSフィールド（プログラムメモリ）620と、PSフィールド620から読出された命令が発火するか否かを示す情報を格納するFCフィールド（待ち合わせメモリ）622を含む。

【0276】図66および図67にそれぞれPSフィールドおよびFCフィールドの1エントリのデータの構成を示す。図66を参照して、PS（プログラム記憶）フィールドは、命令フィールドと行先フィールドを含む。命令フィールドは、命令コードOPCと、この命令コードOPCの命令内容を識別するフラグFLAGを含む。フラグFLAGは、入力データパケット（図64参照）において示すフラグFLAGと同様であり、命令コードOPCが示す演算が2項演算命令、単項演算命令、メモリアクセス命令、およびフィルタ演算命令のいずれであるかを示す。

【0277】この命令コードOPCによる命令の実行後の演算結果が伝達される行先情報は最大2個格納することかできる。第1の行先ノードおよび第2の行先ノード#1および#2はそれぞれ2入力ノードの左右識別フラグL/Rとノード番号ND#およびプロセッサ番号PE

#を含む。行先ノードが2つ存在する場合には、出力時において出力データのコピーを行なうことを示すフラグCOPYが“1”となる。

【0278】図67を参照して、FC（発火制御）フィールドは、相手方のデータを待ち合わせ状態にある待ち合わせオペランドDATA/CONSTと、待ち合わせ状態にある世代を示す待ち合わせ世代フィールドと、各種の制御フラグを含む。待ち合わせオペランドはオペランドデータDATAの場合および定数CONSTの場合がある。待ち合わせ世代フィールドは、世代番号GE#を格納する。フラグ領域には、このFCフィールドにおいて待ち合わせ状態となっているデータが有効であるか否かを示す有効フラグVLDと、行先が2つ存在する場合にコピーすべきことを示すコピーフラグCOPYを含む。

【0279】発火/非発火の判別には、有効フラグVLDが“1”であることを条件として待ち合わせ世代番号GE#の比較が行なわれる。世代番号GE#が一致した場合には発火パケットの生成が行なわれる。すなわち世代番号が一致した場合にはこのFCフィールドの待ち合わせオペランドと入力データパケットのオペランドデータがPSフィールドの命令コードOPCに従って演算実行される。

【0280】命令コードOPCがフィルタ装置510における演算操作を示す場合には、フラグFLAGがフィルタ操作を示す。この場合には、待ち合わせデータは存在しないため、無条件で発火状態とされ、オペランドデータDATA（入力データパケットに含まれるデータ）とこのPSフィールドから読出された命令および行先ノード番号から出力データパケットが生成されてフィルタ装置510へ分岐器586を介して伝達される（図63参照）。

【0281】次に再び図65へ戻って、FCP部584は、キューバッファ582から与えられた命令が実行命令（FCPメモリ600に対するロード/ダンプ命令以外の命令）であるか否かの識別を行なう命令識別部602と、命令識別部602から与えられるデータパケットに含まれるノード番号ND#および世代番号GE#をハッシュアドレスとしてFCPメモリ600をアクセスするアドレスデコーダ604と、アドレスデコーダ604によりアドレス指定されたフィールド（PSフィールドおよびFCフィールド）を読出す命令フェッチ部606と、命令フェッチ部606により読出された命令の種類を識別する第2の命令識別部608と、命令識別部608の識別結果に従ってこのデータパケットの発火/非発火を判定する発火判定部610と、発火判定部610の判定結果に従ってパケットを生成する出力パケット生成部612を含む。次に動作について説明する。

【0282】まずこのFCP部においては以下の動作が実行される。前述のごとく、PSフィールドから読出さ

れた命令の演算結果が伝達されるべきノードが2つある場合には、対応のFCフィールドのフラグCPYが“1”となる。

【0283】また片側の入力が定数データであるオペランドに対しては同一アドレスのFCフィールドに定数データ(CONST)が書込まれかつ定数フラグCST(図67において示す)に“1”が書込まれる。この定数データと対をなすオペランドが入力された場合無条件に発火パケットが生成される。

【0284】また原則として、定数データを持たない2項演算は、左右オペランドのうち先に入力されたオペランドの内容がFCフィールドの待ち合わせオペランド(DATA)に書込まれ、待ち合わせ状態にある有効データであることを示すフラグVLPが“1”とされる。またこのとき世代番号GE#が書込まれる。後から入力されたオペランドは、有効フラグVLDが“1”であることを条件としてその2つのオペランドの内容を参照し、世代番号GE#が一致した場合に発火パケットを生成し、かつ待ち合わせ状態のオペランドが消費されたことを示すために有効性フラグVLDを“0”とする。世代番号GE#が不一致の場合、すなわちハッシュ衝突の場合には、このデータは対を構成しないため、図63においてループを構成するパイプライン(このプロセッサは内部演算はマイクロパイプラインを構成している)上に伝達され、この待ち合わせメモリ622において空き領域が生じるのを待つ。

【0285】さらに、命令フィールドの命令コードOPCがフィルタ操作演算の場合には、FCフィールドにおける内容を参照することなく無条件で発火状態とされ、発火パケットの生成が実行される。次にプログラム実行時におけるFCP部の動作について説明する。

【0286】命令識別部602は、キューバッファ582から与えられたデータパケットに含まれる命令コードOPCのデコードを実行する。この命令コードのデコードは実行命令であるか外部制御命令であるか、FCPメモリに対するロード/ダンプ命令であるかの判別が実行される。実行命令であると判別された場合には、続いて直前のデータパケットに含まれるオペランドと同一アドレスをアクセスするオペランドであるか否かの判定が実行される(ノード番号ND#および世代番号GE#を利用する)。同一アドレス(FCPメモリ600のアドレス)を連続してアクセスするオペランドの場合には、このFCPメモリ600へのアクセスのオーバーヘッドを回避するために、メモリアクセスは実行されない。それに代えて、出力パケット生成部612においてオペランドの対である発火パケットを生成する(この発火パケットの生成手法は事前発火と呼ぶ)。

【0287】アドレスデコーダ604は、次の命令のフェッチのために、この命令識別部602から与えられたデータパケットに含まれるノード番号ND#および世代

番号GE#をハッシュアドレスとしてアドレスデコードを実行し、FCPメモリのアドレスを生成する。ここで、ハッシュアドレスとは、適当な関数(ハッシュ関数)をhとし、FCPメモリ600のアドレスをAとすると、

$$A = h(ND\#, GE\#)$$

で与えられる。

【0288】

$h(ND_i\#, GE_i\#) = h(ND_j\#, GE_j\#)$  の場合、異なる世代番号であってもFCPメモリの同一アドレスAを指定している場合が生じる。この現象はハッシュ衝突と呼ばれる。ハッシュ関数h( )はできるだけ衝突が少なくなるように定義される。ハッシュ関数には様々なものが利用されているが、キーデータ(ノード番号および世代番号)を適当な桁数で分割し、各分割された値を加算する方法(畳み込み方法)がある。この他に、キーを2乗してから中位のビット列を切出す方法がある。この2乗の代わりに割算を用いる場合もある。

【0289】上述のハッシュ演算を実行することにより、FCPメモリ600における内容の検索を高速で実行することができる。

【0290】命令フェッチ部606は、このアドレスデコーダ604が生成したアドレスに従ってFCPメモリ600のPSフィールドおよびFCフィールドを参照する。すなわち、FCPメモリ600におけるPSフィールド620から命令およびこの命令実行後の行先情報を読み出し、またFCフィールド622から片側のオペランドがこのFCPメモリ600のFCフィールド620において待ち合わせを行なっているか(有効フラグVLDを利用)などの情報を読み出す。

【0291】第2の命令識別部608は、PSフィールド620から読出された命令コードOPCの演算内容を識別する。すなわち、図63に示すフィルタ装置510を利用する演算であるか、演算部554を利用する演算であるか、メモリインターフェースを介してのデータメモリへのアクセス命令であるかデータを出力する命令であるかの識別が実行される。

【0292】フィルタ装置510を利用する命令の場合、フラグFLAGにおけるフィルタ演算操作命令を示すフラグに“1”が書込まれている。この場合には、無条件で発火状態とされる。

【0293】演算部554における演算については、単項演算であるか2項演算であるかの識別が実行される。2項演算においてはさらに、定数演算命令であるか、事前発火(命令識別部602において実行された)であるかおよび2変数演算であるかの識別が実行される。

【0294】発火判定部610はこの命令識別部608における命令識別結果に従って発火/非発火を判定する。単項演算命令、定数演算命令、事前発火命令およびフィルタ装置510を利用するフィルタ演算命令の場合

には無条件で発火と判定される。

【0295】発火判定部610は、また、FCフィールドの内容を参照し、有効フラグVLDが“1”であるか否かを判定し、これにより対をなすオペランドがFCPメモリ600の待ち合わせ領域（FCフィールド）で待っているかを判定するとともにさらに世代番号GE#の一致／不一致の判定を行なう。

【0296】2項演算命令において発火対象となるオペランドが待っていないオペランド（FCフィールドにおける有効フラグVLDが“0”）のオペランドについてはFCフィールドへの書込を示すためのフラグが発火判定部610において生成される。発火判定部610は、さらに、FCPメモリ600に対する書込アドレス（ハッシュアドレス）は一致し、有効フラグVLDも“1”であるものの世代番号GE#が不一致のオペランドについてはハッシュ衝突と判定し該判定結果を示すフラグを生成する。

【0297】出力パケット生成部612は、この発火判定部601の判定結果に従って出力パケットを生成する。すなわち、発火と判定されたオペランドに対しては、PSフィールド602から読出された命令コードOPCおよびフラグFLAGおよび行先情報（ノード番号ND#およびプロセッサ番号PE#および2入力識別フラグL/R）で置き換え、かつFCフィールド622から読出された待ち合わせオペランドおよびフラグを追加して発火パケットを生成するとともに対応のFCフィールドの無効化（有効フラグVLDを“0”）をする。

【0298】出力パケット生成部612は、また、無条件発火のデータパケットに対しては、FCフィールド622における変更を行なうことなくPSフィールド620およびFCフィールド622から読出された情報に従ってデータパケットを生成する。

【0299】出力パケット生成部612は、また、無効演算命令において発火対象のオペランドが待ち合わせを行っていない場合には、このデータパケットに含まれるオペランドデータおよび世代番号GE#をFCフィールドへ書込み、かつ有効フラグVLDに対し“1”を書込み、発火対象となるオペランドデータを待ち合わせる状態とする。

【0300】出力パケット生成部612は、ハッシュ衝突時においては、PSフィールドおよびFCフィールド620および622からの情報によるデータパケットの変更を行なうことなくそのまま放出する。

【0301】このFCP部584により生成されたデータパケットは分岐器586（図63）および分岐器588（図63参照）を介して対応の部分へ伝達される。ハッシュ衝突を生じたオペランドデータは、図63に示す構成において分岐器586、分岐器588、および演算部554、合流器580を介してキューバッファ582へ与えられる。このループを待ち合わせメモリ領域に空

きが存在するまで巡回する。このとき演算部554においては演算は実行されない（有効フラグVLDの“0”および“1”により演算実行の可否が識別される）。

【0302】このようなデータ駆動型プロセッサにおいてフィルタ演算命令が実行される場合には、発火判定部610において無条件発火と判定され、出力パケット生成部612において発火パケットが生成され、分岐器586を介してフィルタ装置510へ与えられる。フィルタ装置510はこの分岐器586から与えられたデータパケットに含まれる命令コードOPCに従って、実行されるべきプログラム命令群を選択し所望のフィルタ演算を実行し、該実行結果をこの与えられたデータパケットのオペランドデータ領域に書込み再び合流器580へ伝達する。次にこのフィルタ装置510の構成および動作について説明する。

【0303】図68は、フィルタ装置の構成およびこのフィルタ装置に対する入出力データパケットの構成を示す図である。図68に示すフィルタ装置510は、本質的に図2および図3に示すフィルタ装置と同じ構成を備える。すなわち、フィルタ装置510は、オペランドデータを受ける入力部710と、レジスタファイル714と、乗算器および加算シフト複合演算器を含む演算器群716と、データを出力する出力部718と、演算操作を制御するプログラムコントロール部700と、このプログラムコントロール部700からのプログラムに従ってレジスタファイル714のデータの入出力を制御する書込／読出制御部712を含む。この入力部710、書込／読出制御部712、レジスタファイル714、出力部718および演算器群716の動作は先に図2ないし図62を参照して説明したフィルタ装置のそれと同様である。

【0304】プログラムコントロール部700は、与えられたデータパケットの命令コードをデコードする命令デコーダ752と、命令デコーダ752が特定するフィルタ演算に従って、プログラムメモリ756から対応のプログラム命令群を読出し、該命令の実行を制御するシーケンスコントローラ754を含む。プログラムメモリ756は、その記憶内容が外部からプログラム可能であり、アプリケーションに従ってそのフィルタ演算内容がプログラムされる。この場合、図1に示すデータメモリへのアクセスと同様、プログラムメモリ756へもデータ駆動型プロセッサからの命令に従ってアクセス可能とされ、必要なフィルタ演算命令がプログラムメモリ756に格納される構成が利用されてもよい。

【0305】命令デコーダ752は命令コードOPCをデコードし、実行されるべきフィルタ演算を特定するが、この命令デコーダ752はまたシーケンスコントローラ754内に含まれてもよい。命令デコーダ752は、レジスタファイル714に設定されたフィルタ演算内容と現在与えられた命令コードが示すフィルタ演算が

47

同じ場合、その旨をシーケンスコントローラ 754 へ知らせる。シーケンスコントローラ 754 はこのとき、プログラムメモリ 756 へはアクセスせず入力データの書込および出力データの出力を実行する。次に簡単に動作について説明する。

【0306】データ駆動型プロセッサからフィルタ演算を実行すべきデータパッケージが与えられると、このデータパッケージのうち命令コード OPC # 1 が命令デコーダ 752 へ与えられる。いま、命令デコーダ 752 が命令コード OPC # 1 をデコードし、フィルタ演算 FIRFUNC 4 が指定されたと判定された状態を想定する。このフィルタ演算 FIRFUNC 4 は、前述の 4 次の FIR 命令であるとする。このときシーケンスコントローラ 754 はプログラムメモリ 756 から先に説明したような 4 次の FIR 命令 (fir 命令群) を読出し、書込/読出制御部 712 を介して必要なデータをレジスタファイル 714 に設定する。

【0307】この設定の後、データパッケージに含まれたオペランドデータ DATA (入力データ in) が入力部 710 および書込/読出制御部 712 を介してレジスタファイル 714 へ与えられ指定された 4 次の FIR 演算が実行される。この実行結果は出力部 718 を介して出力される。出力部 718 から出力されるデータパッケージ

48

は命令コードおよび行先情報は入力データパッケージのそれと同じであるが、オペランドデータが出力データ DATA (out) に変更される。この出力データパッケージは図 63 に示す合流器 580 へ与えられる。

【0308】ここで図 68 においては、データパッケージがそれぞれ命令デコーダおよび入力部へ与えられる構成が示されている。データバスと命令バスとを分離することにより、これは実現されるが、入出力インターフェース回路が設けられ、入力インターフェース部において命令コードおよびオペランドデータの振り分けが実行され、また出力インターフェース部において、このデータパッケージにおけるオペランドデータの書込が実行され、最終的な出力データパッケージが生成される構成が利用されてもよい。

【0309】ここで、図 68 において、フラグ F・FLAG がフィルタ演算命令であることを示すフラグである。

【0310】データ駆動型フィルタ装置とデータ駆動型プロセッサとを組合せたプログラムの例を以下に示す。このプログラムは C 言語で記述される。

【0311】

【数 1】

```

49
# define OFFSET 5      "OFFSETに値5を定義する。"
main(in)
{
    int in, out;        "パラメータおよび"
    int a = 0;          "その型の定義"

    static int b[10];   "配列の領域の"
                        "確保"

    {
        FIRFUNC4(in, out); "関数呼出しの形式"
                           "でフィルタ演算を呼出す"

        a = out * OFFSET + a; "フィルタ演算の結果に"
                              "OFFSETを掛けたaを"
                              "加算する"

        b[5] = a;          "配列bの5番目にaを"
                           "書込む"
    }
}

```

【0312】このプログラム例においては、OFFSETに値5が定義され、次いでmain関数の読出が行なわれ、このmain関数に従って処理が実行される。まず引数（パラメータ）in, outおよびaがその型とともに定義され、次いで配列領域が確保される。ここでは配列は固定的に10個準備される。次いで関数呼出しの形式でフィルタ演算が読出される。このフィルタ演算はFIRFUNC4(in, out)である。この命令が指定された場合には、フィルタ演算がフィルタ装置部において実行される。

【0313】次に演算 $a = out * OFFSET + a$ の演算が実行される。この演算はデータ駆動型プロセッサの演算部554（図63参照）で実行される。この実行結果aは配列bの第5番目に書込まれる。この配列bは図63に示すデータメモリにその領域が確保されており、このデータメモリへのアクセスによりデータの書込が実行される。

【0314】このプログラムを実行した場合、まず図63における合流器580およびキューバッファ582を介してFCP部584へ入力データが与えられる。FC

P部584においては、その行先情報に従ってフィルタ演算命令FIRFUNC4(in, out)が読出される。この命令はフィルタ演算命令であるため無条件発火とされ、図63に示す分岐器586を介してフィルタ装置510へ与えられる。このフィルタ装置510へはフィルタ演算命令FIRFUNC4(in, out)とともにオペランドデータが与えられる。したがってフィルタ演算が施されるべき入力データはすべてフィルタ装置510へ送出され、そこで指定されたフィルタ演算が実行される。

【0315】この演算結果は、再びデータパケットの形態で図63に示す合流器580およびキューバッファ582を介してFCP部584へ伝達される。FCP部584においては、この与えられたデータパケットの行先情報から演算命令を読出し、乗算 $out * OFFSET$ を実行するとともにデータメモリ570へアクセスし、配列bの5番目b[5]のデータを読出し、加算し、この加算結果を再び配列bの5番目に書込む。この一連の演算はデータ駆動型プロセッサにおける演算であり、フィルタ装置におけるフィルタ演算と独立に図63に示す

演算部554において実行される。配列bの5番目にaを書込む演算命令はメモリインターフェース560を介してデータメモリへアクセスすることにより実行される。

【0316】上述の一連の動作により、その内容にフィルタ演算処理を含む複雑なプログラムであってもデータ駆動方式に従ってすべての演算命令を実行することができ、効率的かつ高速で複雑なプログラムであっても実行することが可能となる。

【0317】なお上記実施例においては、データ駆動型プロセッサとしては上述の構成に限定されず、データ駆動方式に従って処理を実行するものであれば任意の構造のプロセッサを利用することができる。

【0318】〔処理性能〕本発明によるデータ駆動型情報処理装置を用いた場合、従来のデータ駆動型プロセッサのみを用いるフィルタ処理と比べて、IIRフィルタ処理を実現する場合には45ないし60倍の処理速度が得られる。同様にデータ駆動型プロセッサのみを用いてFIRフィルタ処理を実現する場合の構成と比べて5ないし10倍の処理速度を得ることができる。

【0319】また専用LSIを用いるフィルタ装置と比べて、タップ数を自由に設定することができる（レジスタファイルにおけるエントリーの数の増減はプログラムにより容易に実行することができる）。このため、フィルタ装置を処理用途に応じて任意にその構成を設定することができ、いずれの用途においても容易に対応することが可能となる。

【0320】さらに、上述の実施例においては、IIRフィルタおよびFIRフィルタそれぞれを個別に構成しているが、両者を組み合わせる結合型フィルタをも容易に構成することが可能となる。

【0321】また、データ駆動型フィルタ装置とデータ駆動型プロセッサとを組合せて利用するため、たとえば高速フーリエ変換を含む画像データ処理などのようなフィルタ処理を含む複雑なアプリケーションにも容易に対応することができる。またこのときデータ駆動型フィルタ装置およびデータ駆動型プロセッサともにデータ駆動方式で処理を実行しているため、必要とされるプログラムの形式が同じであり、異なる形式のプログラムを用いる必要がなく、装置間の整合性がよく、またプログラム作成が容易となる。

#### 【0322】

【発明の効果】以上のように、この発明によれば、データ駆動方式に従ってフィルタ操作を実行するデータ駆動型フィルタ装置と、データ駆動方式に従って演算命令を実行するデータ駆動型プロセッサを組合せてデータ駆動型情報処理装置を構成したため、フィルタ操作をその内容に含む複雑なアプリケーションであっても容易に対応することが可能となる。またそのときデータ駆動型フィルタ装置が実行するフィルタ操作がプログラム可能であ

るため、任意のフィルタ操作を容易に実現することが可能となり、種々のアプリケーションに対して柔軟に対処することのできるデータ駆動型情報処理を得ることができる。

【0323】さらに、データ、定数および初期値が単一代入単一読出のルールで書込／読出が行なわれる複数のレジスタを有するレジスタファイルと、乗算器と加算シフト複合演算器とを有する演算器群とでフィルタ装置を構成したため、FIRフィルタ処理およびIIRフィルタ処理における基本フィルタ演算を最小命令単位として表現することができ、フィルタ装置においては、フィルタ処理にのみ関連する命令を用いてフィルタ処理を実行することができ、高速でフィルタ処理を実行することが可能となる。

【0324】また、レジスタファイルのエントリー数はタップ数に対応するため、このエントリー数はプログラムにより設定可能であり、さらに任意の構成のフィルタを容易に実現することが可能となる。

#### 【図面の簡単な説明】

【図1】この発明の一実施例であるデータ駆動型情報処理装置の構成を示す図である。

【図2】この発明に従うデータ駆動型フィルタ装置の全体の構成を概略的に示すブロック図である。

【図3】図2に示すレジスタファイルの概略構成を示す図である。

【図4】iir命令におけるレジスタファイル内のレジスタ間の接続およびレジスタファイル内のレジスタの構成を示す図である。

【図5】fir命令におけるレジスタファイル内のレジスタ間の接続およびレジスタファイルの構成を示す図である。

【図6】iir命令およびfir命令両者兼用時におけるレジスタファイルの接続構成およびレジスタファイルの構成を示す図である。

【図7】iir命令の記述形式およびその内部構造を示す図である。

【図8】fir命令の記述形式およびその内部構造を示す図である。

【図9】2次のIIRフィルタの構成と本発明におけるiir命令への分解手順を示す図である。

【図10】図9に示す2次のIIRフィルタをiir命令で記述した際のフローグラフを示す図である。

【図11】図10に示すフローグラフをテキスト形式で記述した際のプログラムコーディングを示す図である。

【図12】図9に示すフィルタ処理を実現する際の初期状態におけるレジスタファイルの各レジスタの格納データの状態および演算器との接続構成を示す図である。

【図13】IIRフィルタ処理実行時における実行サイクル1におけるレジスタファイル内の各レジスタの格納データおよびそのときのデジタルフィルタにおけるデー

タの分布を示す図である。

【図14】IIRフィルタ処理実行時における第2実行サイクルにおける各レジスタの格納データおよびそのときのデジタルフィルタのデータの分布を示す図である。

【図15】IIRフィルタ処理実行時における第3実行サイクルにおける各レジスタの格納データおよびそのときのデジタルフィルタのデータの分布を示す図である。

【図16】IIRフィルタ処理実行時における第4実行サイクルにおける各レジスタの格納データおよびそのときのデジタルフィルタのデータの分布を示す図である。

【図17】IIRフィルタ処理実行時における第5実行サイクルの動作を詳細に示す図である。

【図18】IIRフィルタ処理実行時における第5実行サイクルの詳細な動作およびこのサイクル完了時における各レジスタの格納データおよびそのときのデジタルフィルタにおけるデータの分布を示す図である。

【図19】IIRフィルタ処理実行時における第6実行サイクル時の各レジスタの格納データおよびそのときのデジタルフィルタにおけるデータの分布を示す図である。

【図20】IIRフィルタ処理実行時における第7実行サイクル時の各レジスタの格納データおよびそのときのデジタルフィルタにおけるデータの分布を示す図である。

【図21】IIRフィルタ処理実行時における第8実行サイクル時の各レジスタの格納データおよびそのときのデジタルフィルタにおけるデータの分布を示す図である。

【図22】IIRフィルタ処理実行時における第9実行サイクル時の各レジスタの格納データを示す図である。

【図23】IIRフィルタ処理実行時における第10実行サイクル時の各レジスタの格納データを示す図である。

【図24】IIRフィルタ処理実行時における第11実行サイクル時の各レジスタの格納データを示す図である。

【図25】IIRフィルタ処理実行時における第12実行サイクル時の各レジスタの格納データを示す図である。

【図26】IIRフィルタ処理実行時における第13実行サイクル時の各レジスタの格納データを示す図である。

【図27】IIRフィルタ処理実行時における第14実行サイクル時の各レジスタの格納データを示す図である。

【図28】IIRフィルタ処理実行時における第15および16実行サイクル時の各レジスタの格納データを示す図である。

【図29】IIRフィルタ処理実行時の第2ないし第16動作サイクル時における各演算器の稼働状態を示す図

である。

【図30】複数の演算器を用いたIIRフィルタ処理実行時における各レジスタの格納データおよびそのときのデジタルフィルタにおけるデータの分布を示す図である。

【図31】複数の演算器を用いてIIRフィルタ処理を実行する際の第1実行動作サイクル時における各レジスタの格納データおよびそのときのデジタルフィルタ内のデータの分布を示す図である。

【図32】複数の演算器を用いてIIRフィルタ処理を実行する際の第2実行サイクル時の格納データおよびそのときのデジタルフィルタ内のデータの分布を示す図である。

【図33】複数の演算器を用いてIIRフィルタ処理を実行する際の第3の動作サイクル時における各レジスタの格納データおよびそのときのデジタルフィルタ内のデータの分布を示す図である。

【図34】複数の演算器を用いてIIRフィルタ処理を実行する際の乗算操作完了時における各レジスタの格納データおよびそのときのデジタルフィルタ内のデータの分布を示す図である。

【図35】複数の演算器を用いてIIRフィルタ処理を実行する際の第3の動作サイクル時における各レジスタの格納データおよびそのときのデジタルフィルタ内のデータの分布を示す図である。

【図36】複数の演算器を用いてIIRフィルタ処理を実行する際の第4の動作サイクル時における各レジスタの格納データおよびそのときのデジタルフィルタ内のデータの分布を示す図である。

【図37】複数の演算器を用いてIIRフィルタ処理を実行した際の各動作サイクル時における演算器の稼働状況を示す図である。

【図38】4次のFIRフィルタの構成およびfir命令への分解手順を示す図である。

【図39】図38に示す4次のFIRフィルタをfir命令で記述する際のフローグラフを示す図である。

【図40】図39に示すフローグラフをテキスト形式で記述した際のプログラムコーディングを示す図である。

【図41】FIRフィルタ処理実行時における初期状態の各レジスタ格納データおよび演算器との接続関係とそのときのデジタルフィルタ内のデータの分布を示す図である。

【図42】FIRフィルタ処理実行時における第1実行サイクル時における各レジスタの格納データおよびそのときのデジタルフィルタ内のデータの分布を示す図である。

【図43】FIRフィルタ処理実行時における第2実行サイクル時における各レジスタの格納データおよびそのときのデジタルフィルタ内のデータの分布を示す図である。



【図 4 4】FIR フィルタ処理実行時における第 3 実行サイクル時における各レジスタの格納データおよびそのときのデジタルフィルタ内のデータの分布を示す図である。

【図 4 5】FIR フィルタ処理実行時における第 4 実行サイクル時における各レジスタの格納データおよびそのときのデジタルフィルタ内のデータの分布を示す図である。

【図 4 6】FIR フィルタ処理実行時における第 5 実行サイクル時における各レジスタの格納データおよびそのときのデジタルフィルタ内のデータの分布を示す図である。

【図 4 7】FIR フィルタ処理実行時における第 6 実行サイクル時における各レジスタの格納データおよびそのときのデジタルフィルタ内のデータの分布を示す図である。

【図 4 8】FIR フィルタ処理実行時における第 7 実行サイクル時における各レジスタの格納データおよびそのときのデジタルフィルタ内のデータの分布を示す図である。

【図 4 9】FIR フィルタ処理実行時における第 8 実行サイクル時における各レジスタの格納データおよびそのときのデジタルフィルタ内のデータの分布を示す図である。

【図 5 0】FIR フィルタ処理実行時における第 9 実行サイクル時における各レジスタの格納データおよびそのときのデジタルフィルタ内のデータの分布を示す図である。

【図 5 1】FIR フィルタ処理実行時における第 10 実行サイクル時における各レジスタの格納データおよびそのときのデジタルフィルタ内のデータの分布を示す図である。

【図 5 2】FIR フィルタ処理実行時における第 11 実行サイクル時における各レジスタの格納データおよびそのときのデジタルフィルタ内のデータの分布を示す図である。

【図 5 3】FIR フィルタ処理実行時における第 12 実行サイクル時における各レジスタの格納データおよびそのときのデジタルフィルタ内のデータの分布を示す図である。

【図 5 4】FIR フィルタ処理実行時における第 13 実行サイクル時における各レジスタの格納データおよびそのときのデジタルフィルタ内のデータの分布を示す図である。

【図 5 5】FIR フィルタ処理実行時における第 14 実行サイクル時における各レジスタの格納データおよびそのときのデジタルフィルタ内のデータの分布を示す図である。

【図 5 6】FIR フィルタ処理実行時における第 15 実行サイクル時における各レジスタの格納データおよびそ

のときのデジタルフィルタ内のデータの分布を示す図である。

【図 5 7】FIR フィルタ処理実行時における第 2 ないし第 16 実行サイクル時における各演算器の稼働状況を示す図である。

【図 5 8】複数の演算器を用いた FIR フィルタ処理実行時における各レジスタの格納データを示す図である。

【図 5 9】複数の演算器を用いて FIR フィルタ処理を実行する際の第 1 動作サイクル時における各レジスタの格納データおよびそのときのデジタルフィルタ内のデータの分布を示す図である。

【図 6 0】複数の演算器を用いて FIR フィルタ処理を実行する際の第 2 動作サイクル時における各レジスタの格納データおよびそのときのデジタルフィルタ内のデータの分布を示す図である。

【図 6 1】レジスタファイル内のレジスタの読出制御を行なう構成の一例を示す図である。

【図 6 2】図 6 1 に示す読出制御回路の構成および動作を示す図である。

【図 6 3】図 1 に示すプログラム／待ち合わせメモリ部の具体的構成を示す図である。

【図 6 4】データ駆動型プロセッサへ与えられるデータパケットの構成を示す図である。

【図 6 5】図 6 3 に示す FCP 部の具体的構成を示す図である。

【図 6 6】図 6 5 に示す PS フィールドの 1 エントリの構成を示す図である。

【図 6 7】図 6 5 に示す FC フィールドの 1 エントリの構成を示す図である。

【図 6 8】この発明の一実施例において利用されるデータ駆動型フィルタ装置の構成およびこのフィルタ装置に対する入出力データパケットの構成を示す図である。

#### 【符号の説明】

- 1, 1a ~ 1n 乗算器
- 2, 2a ~ 2n 加算シフト複合演算器
- 10 レジスタファイル
- 20 入力部
- 22 加算器
- 24 シフト演算器
- 30 出力部
- 40 第 1 の演算器群
- 50 第 2 の演算器群
- 500 データ駆動型情報処理装置
- 510 データ駆動型フィルタ装置
- 550 データ駆動型プロセッサ
- 552 プログラム／待ち合わせメモリ部
- 554 演算部
- 556 入力部
- 558 出力部
- 560 メモリインターフェース (I/F)

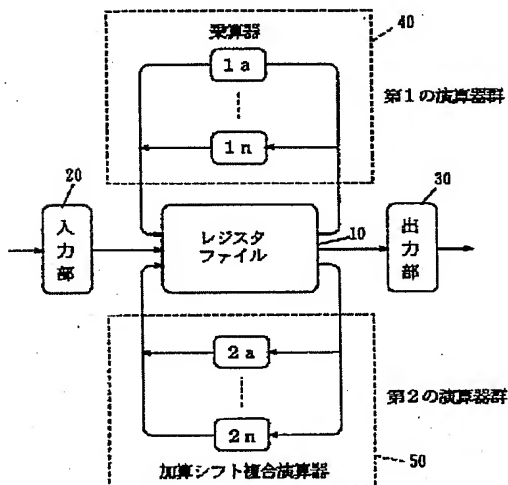
57

570 データメモリ  
 580 FCP部  
 600 FCPメモリ  
 602 命令識別部  
 604 アドレスデコーダ  
 606 命令フェッチ部  
 608 命令識別部  
 610 発火判定部  
 612 出力パケット生成部  
 620 PSフィールド(プログラムメモリ)  
 622 FCフィールド(待ち合わせメモリ)  
 700 プログラムコントロール部  
 710 入力部  
 712 書込/読出制御部  
 714 レジスタファイル  
 716 演算器群

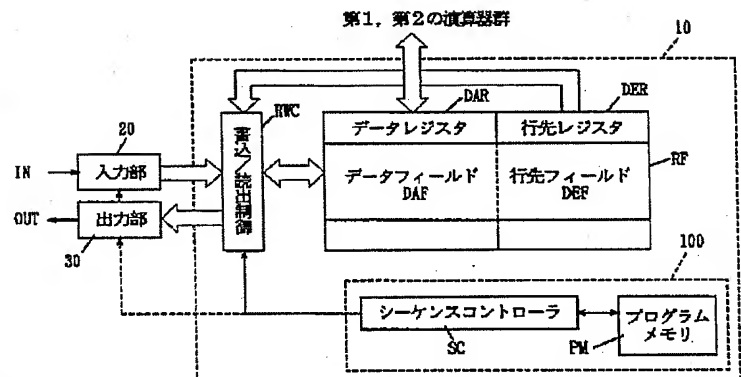
58

718 出力部  
 752 命令デコーダ  
 754 シーケンスコントローラ  
 DAF データフィールド  
 DEF 行先フィールド  
 DAR データレジスタ  
 DER 行先レジスタ  
 A1 レジスタ  
 A2 レジスタ  
 10 C レジスタ  
 B レジスタ  
 X レジスタ  
 D レジスタ  
 E レジスタ  
 F レジスタ

【図2】

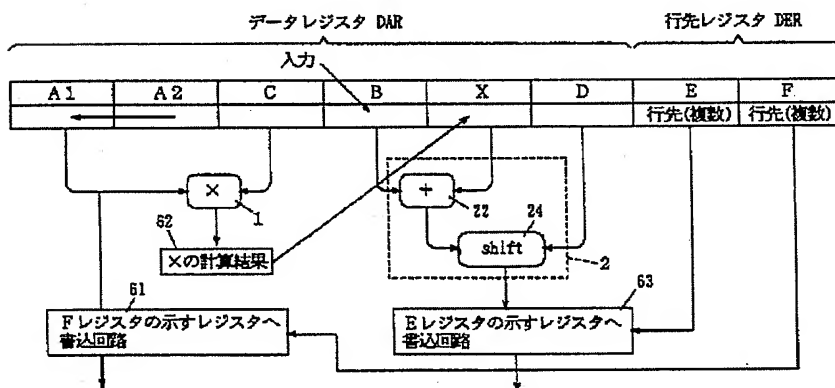


【図3】



【図66】

【図4】



IIR命令によるレジスタ間の接続

PSフィールド

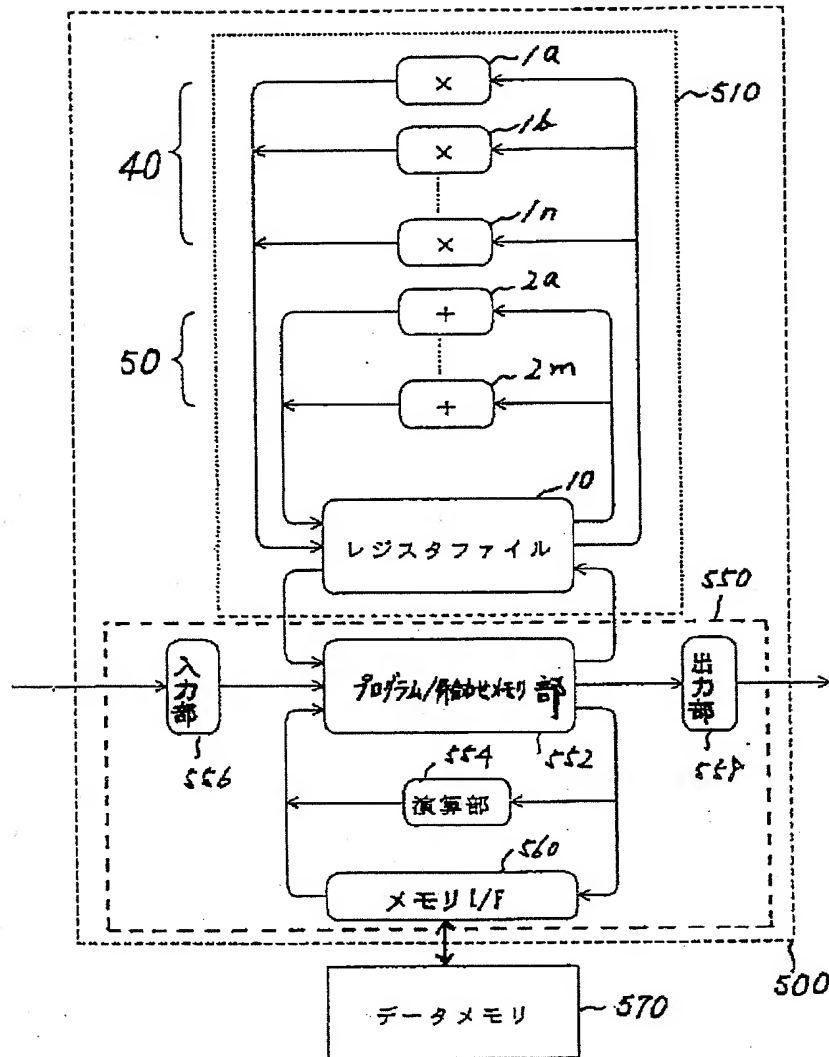
命令	行先#1	行先#2
OPC	ND# PE#	ND# PE#

【図67】

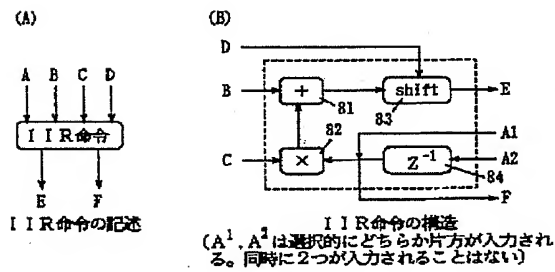
待合せ オペランド	待合せ世代	Z C P D Y
DATA/CONST	GE #	*

FCフィールド

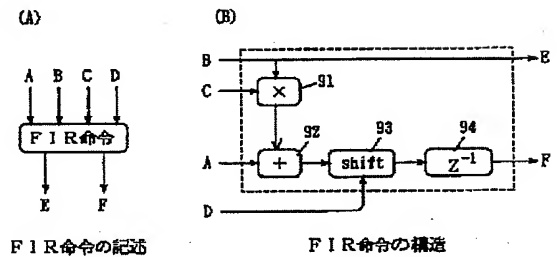
【図1】



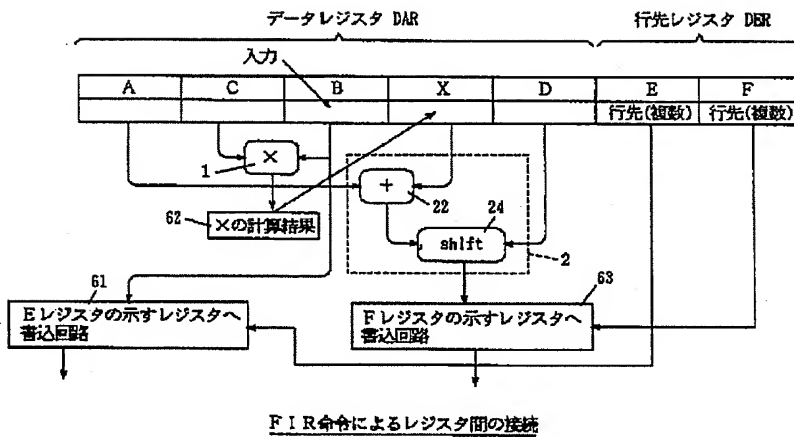
【図7】



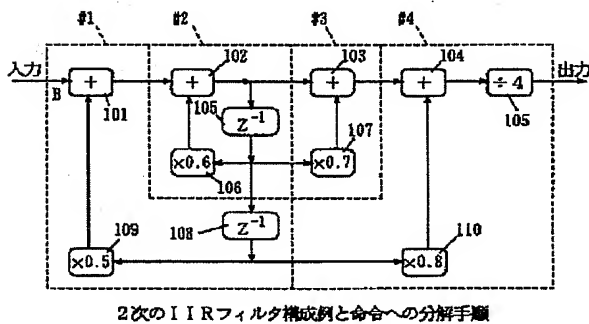
【図8】



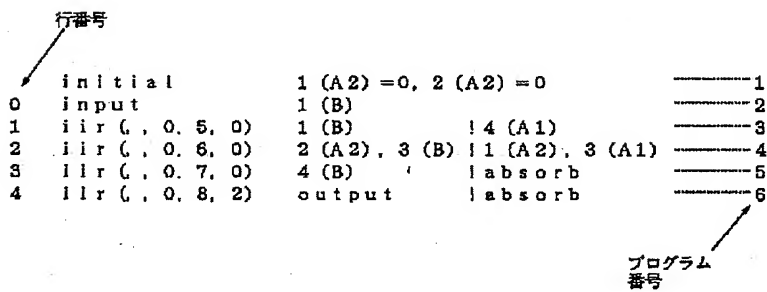
【図5】



【図9】



【図11】



【図22】

実行動作9

	A1	A2	C	B	X	D
1	0	0	0.5	2	0	0
2	0	1.5	0.6	0	0	0
3	0	使用せず	0.7	1.5	0	0
4	0	使用せず	0.8	0	0	2

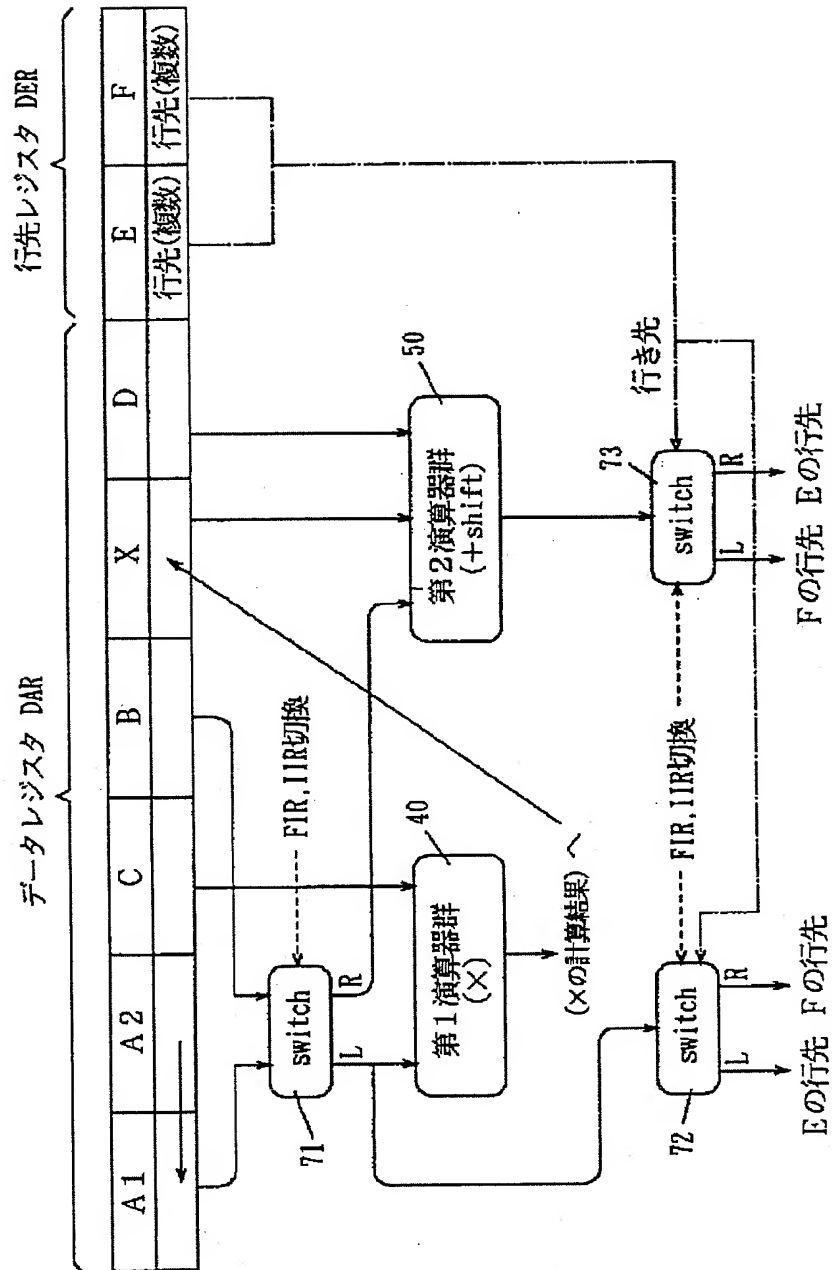
【図23】

実行動作10

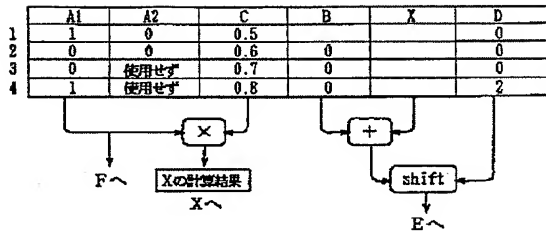
	A1	A2	C	B	X	D
1	0	0	0.5	2	0	0
2	0	1.5	0.6	0	0	0
3	0	使用せず	0.7	1.5	0	0
4	0	使用せず	0.8	0	0	2

出力

【図 6】

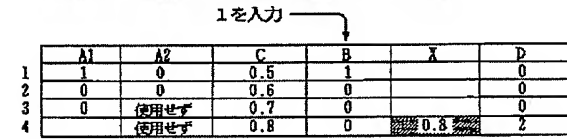


【図12】

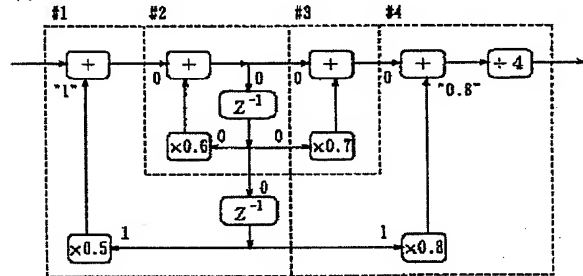


【図13】

(A) 実行動作1 (斜線部分が演算器の稼働部分、以下同様)

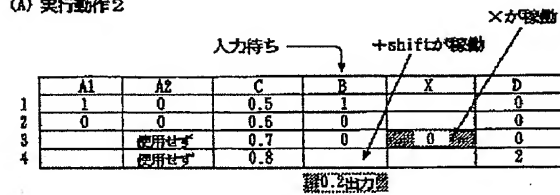


(B)

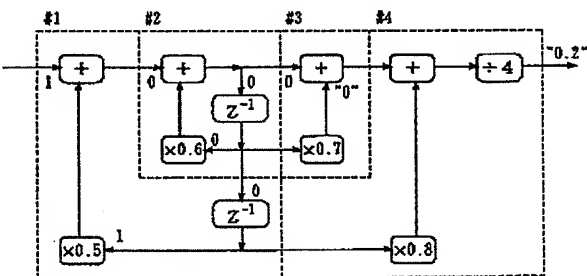


【図14】

(A) 実行動作2

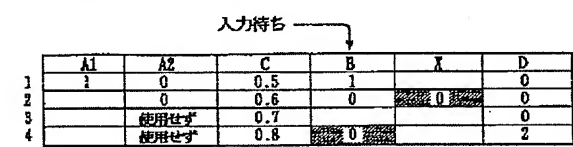


(B)

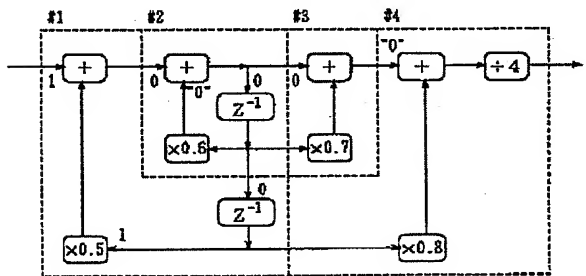


【図15】

(A) 実行動作3

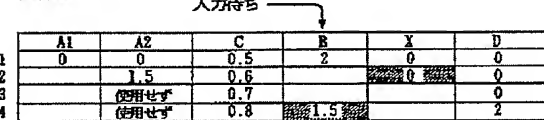


(B)



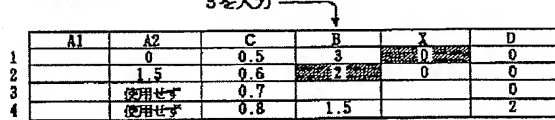
【図24】

実行動作11



【図25】

実行動作12



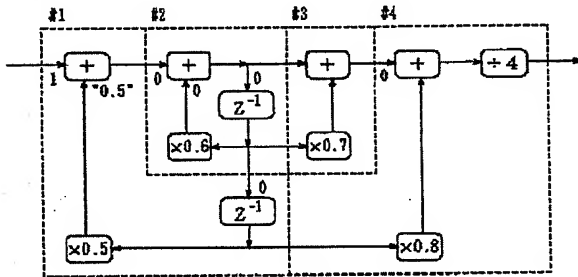
【図16】

(A) 実行動作4

入力待ち

	A1	A2	C	B	X	D
1	0	0	0.5	1	0.5	0
2	0	0	0.6	0	0	0
3		使用せず	0.7			0
4		使用せず	0.8	0		2

(B)



【図18】

(A)

	A1	A2	C	B	X	D
1	0	0	0.5	1	0.5	0
2	0	0	0.6			0
3	0	使用せず	0.7	0		0
4		使用せず	0.8	0	0	2

×

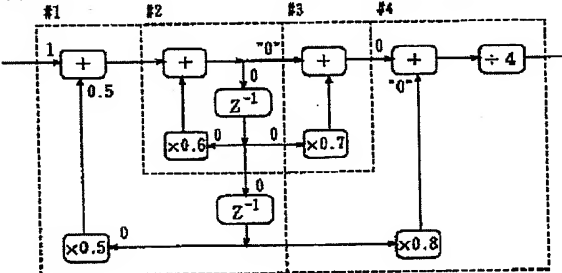
(B)

最終的な状態は以下のようになる。

入力待ち

	A1	A2	C	B	X	D
1	0	0	0.5	1	0.5	0
2	0	0	0.6			0
3	0	使用せず	0.7	0	0	0
4		使用せず	0.8	0	0	2

(C)



【図17】

(A) 実行動作5

より詳細な動き (1)  
レジスタ1 (A2) の値がレジスタ1 (A1)、レジスタ4 (A1) にコピーされる。

	A1	A2	C	B	X	D
1	0	0	0.5	1	0.5	0
2	0	0	0.6	0	0	0
3	0	使用せず	0.7			0
4	0	使用せず	0.8	0		2

(B)

より詳細な動き (2)  
レジスタ2 (A2) の値がレジスタ1 (A2)、レジスタ2 (A1)、レジスタ3 (A1) にコピーされる。

	A1	A2	C	B	X	D
1	0	0	0.5	1	0.5	0
2	0	0	0.6	0	0	0
3	0	使用せず	0.7			0
4	0	使用せず	0.8	0		2

(C)

より詳細な動き (3)  
レジスタ2 (B) の値とレジスタ2 (Xの計算結果) の加算shiftが行われ、レジスタ2 (A2)、レジスタ3 (B) にコピーされる。

	A1	A2	C	B	X	D
1	0	0	0.5	1	0.5	0
2	0	0	0.6			0
3	0	使用せず	0.7	0		0
4	0	使用せず	0.8	0		2

+shift

【図19】

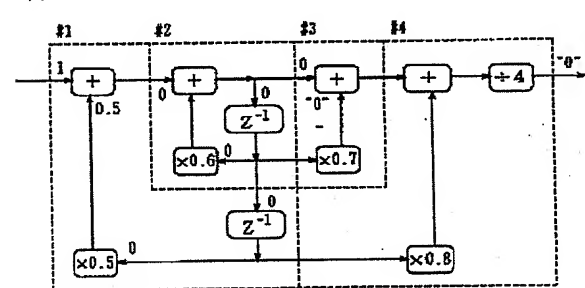
(A) 実行動作6

入力待ち

	A1	A2	C	B	X	D
1	0	0	0.5	1	0.5	0
2	0	0	0.6			0
3	0	使用せず	0.7	0	0	0
4		使用せず	0.8	0		2

0出力

(B)



【図26】

実行動作13

入力待ち

	A1	A2	C	B	X	D
1	0	1.5	0.5	3	0	0
2	1.5	2	0.6			0
3	1.5	使用せず	0.7	2		0
4		使用せず	0.8	1.5	0	2



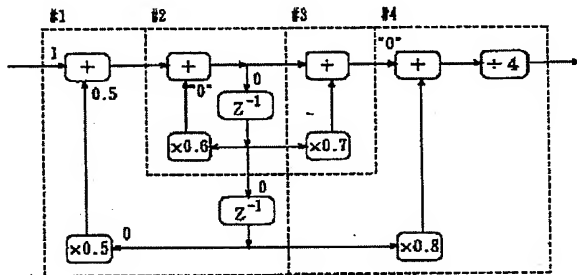
【図20】

(A) 実行動作7

入力待ち

	A1	A2	C	B	X	D
1	0	0	0.5	1	0.5	0
2		0	0.6		0.5	0
3		使用せず	0.7			0
4		使用せず	0.8	0		2

(B)



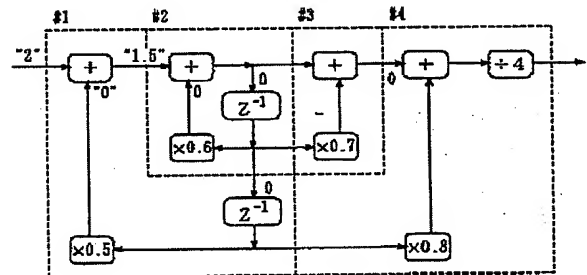
【図21】

(A) 実行動作8

2を入力

	A1	A2	C	B	X	D
1		0	0.5	2	0	0
2		0	0.6	1.5	0	0
3		使用せず	0.7			0
4		使用せず	0.8	0		2

(B)



【図27】

実行動作14

入力待ち

	A1	A2	C	B	X	D
1	0	1.5	0.5	3	0	0
2	1.5	2	0.6		0.9	0
3		使用せず	0.7	2	1.05	0
4		使用せず	0.8			2

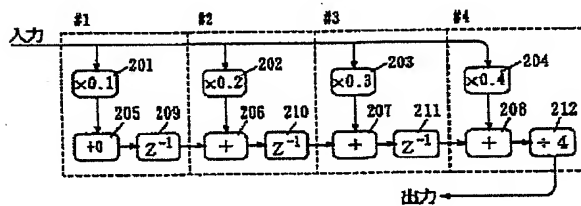
約3.38出力

【図29】

サイクル2～16の各演算器の稼働状況  
(○は稼働状態、空白がデータ待ち状態)

実行動作番号	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
データ入力	○															
x (乗算器)		○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
+ shift (複合演算器)	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
データ出力																○

【図38】



【図28】

(A) 実行動作15

入力待ち

	A1	A2	C	B	X	D
1	0	1.5	0.5	3	0	0
2		2	0.6		0.9	0
3		使用せず	0.7			0
4		使用せず	0.8	3.05		2

(B) 実行動作16

4を入力

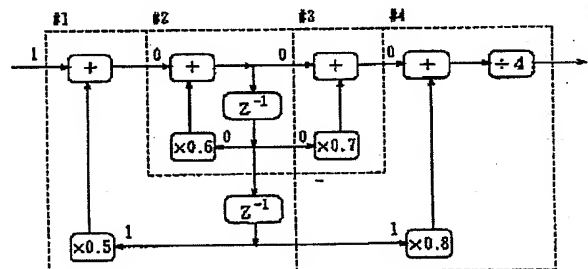
	A1	A2	C	B	X	D
1		1.5	0.5	4	0	0
2		2	0.6	3	0.9	0
3		使用せず	0.7			0
4		使用せず	0.8	3.05		2

【図30】

(A) 初期状態

	A1	A2	C	B	X	D
1	1	0	0.5	1		0
2	0	0	0.6	0		0
3	0	使用せず	0.7	0		0
4	1	使用せず	0.8	0		2

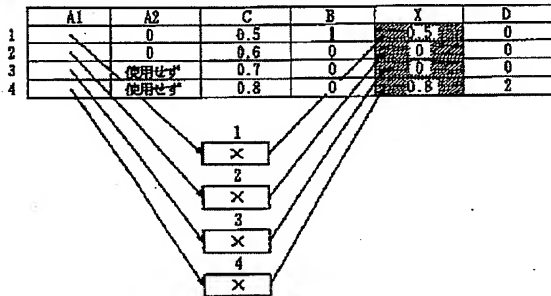
(B)



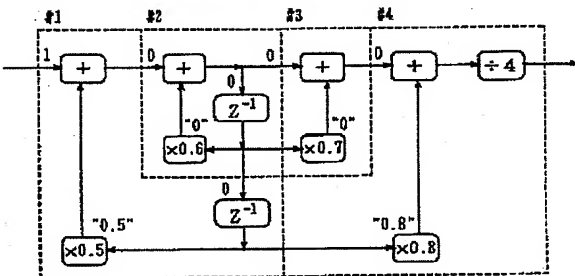
【図31】

(A) 実行状態1-b (1-aは演算器の動作なし)

A1とCの値を4つの×演算器に送り、演算結果を所定のレジスタに書き込む。



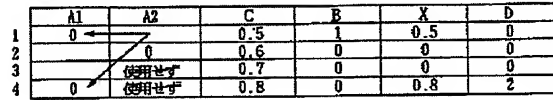
(B)



【図32】

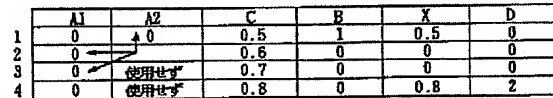
(A) 実行状態2-a

レジスタ1のA2の値が、レジスタ1 (A1)、レジスタ4 (A1)に書き込まれる。

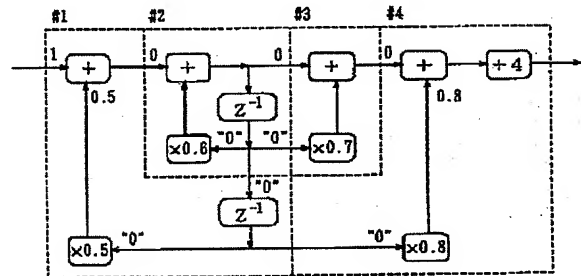


(B)

レジスタ2のA2の値が、レジスタ1 (A2)、レジスタ2、3 (A1)に書き込まれる。



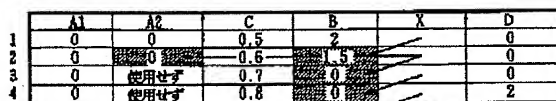
(C)



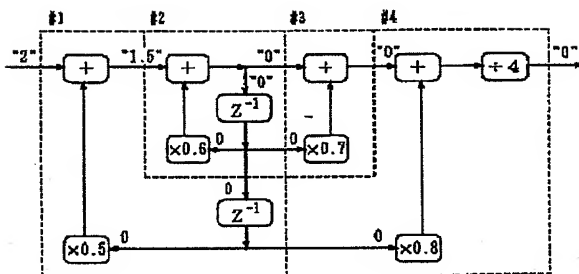
【図33】

(A)

(Xの計算結果)とBの値を+shift演算器に送り、その結果をEレジスタの示す行き先に書き込む (この場合は次のレジスタ番号、すなわち自身が1ならば2、2ならば3というように次のレジスタのBと、2のみA2)

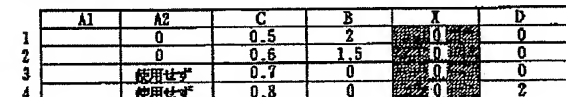


(B)

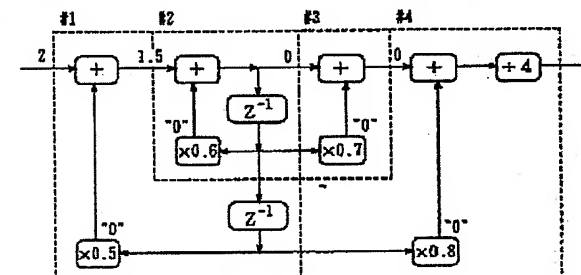


【図34】

(A) 実行状態2-b



(B)



【図35】

(A) 実行状態3-a

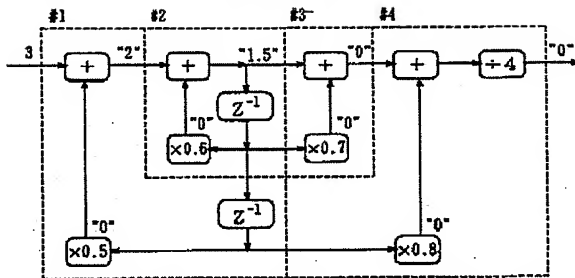
	A1	A2	C	B	X	D
1	0	0	0.5	3		0
2	0	1.5	0.6	2		0
3	0	使用せず	0.7	1.5		0
4	0	使用せず	0.8	0		2

出力7.0

(B) 実行状態3-b

	A1	A2	C	B	X	D
1		0	0.5	3	0.8	0
2		1.5	0.6	2	0	0
3		使用せず	0.7	1.5	0	0
4		使用せず	0.8	0	0	2

(C)



【図37】

実行動作番号	1	2	3	4	
データ入力	○	○	○	○	
×	○	○	○	○	演算器1
+shift(複合演算器)	○	○	○	○	(×、+shift複合演算器1)
×	○	○	○	○	演算器2
+shift(複合演算器)	○	○	○	○	(×、+shift複合演算器2)
×	○	○	○	○	演算器3
+shift(複合演算器)	○	○	○	○	(×、+shift複合演算器3)
×	○	○	○	○	演算器4
+shift(複合演算器)	○	○	○	○	(×、+shift複合演算器4)
データ出力	○	○	○	○	

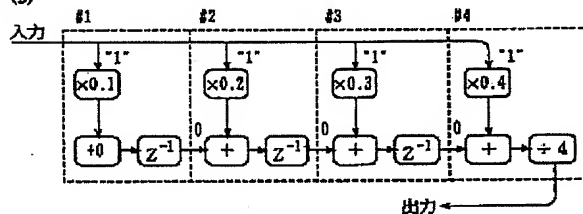
【図42】

(A) 実行状態1

1を入力

	B	C	A	X	D
1	1	0.1	0		0
2	1	0.2	0		0
3	1	0.3	0		0
4	1	0.4	0		2

(B)



【図36】

(A) 実行状態4-a

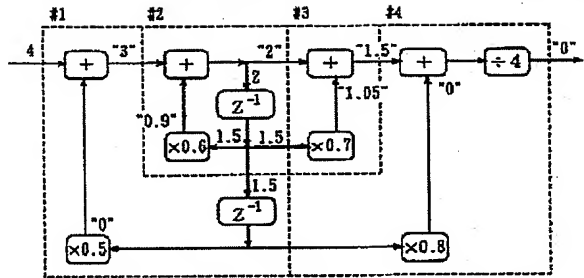
	A1	A2	C	B	X	D
1	0	1.5	0.5	4		0
2	1.5	2	0.6	3		0
3	1.5	使用せず	0.7	2		0
4	0	使用せず	0.8	1.5		2

出力10

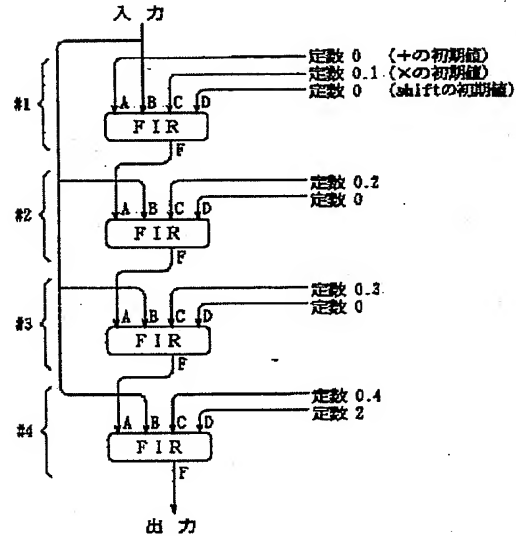
(B) 実行状態4-b

	A1	A2	C	B	X	D
1		1.5	0.5	4	0.9	0
2		2	0.6	3	0.9	0
3		使用せず	0.7	2	1.05	0
4		使用せず	0.8	1.5	0	2

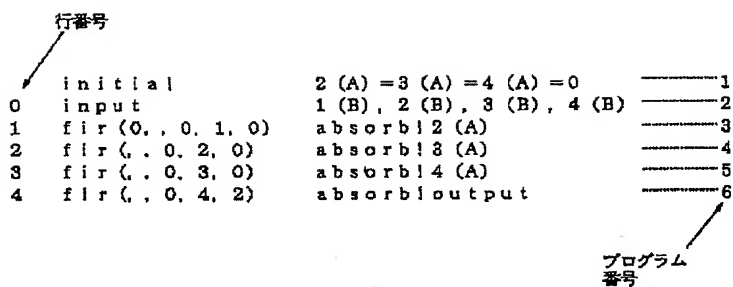
(C)



【図39】



【図40】



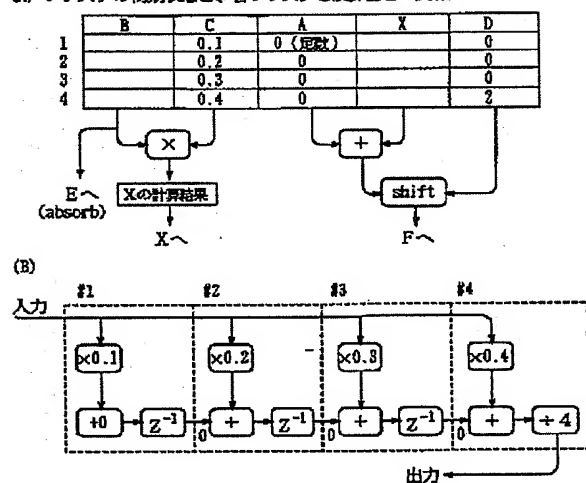
【図58】

初期状態

	B	C	A	X	D
1	1	0.1	0		0
2	1	0.2	0		0
3	1	0.3	0		0
4	1	0.4	0		2

【図41】

(A) レジスタの初期状態と、各レジスタと演算器との関係

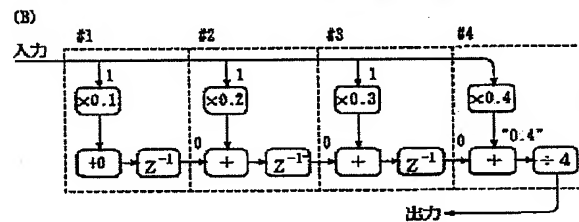


【図43】

(A) 実行状態2

入力待ち

	B	C	A	X	D
1	1	0.1	0		0
2	1	0.2	0		0
3	1	0.3	0		0
4		0.4	0	0.4	2

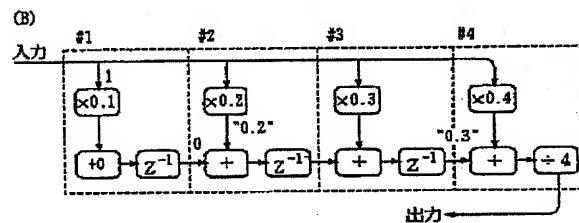


【図45】

(A) 実行状態4

入力待ち

	B	C	A	X	D
1	1	0.1	0		0
2		0.2	0	0.2	0
3		0.3	0		0
4		0.4	0.3		2



【図44】

(A) 実行状態3

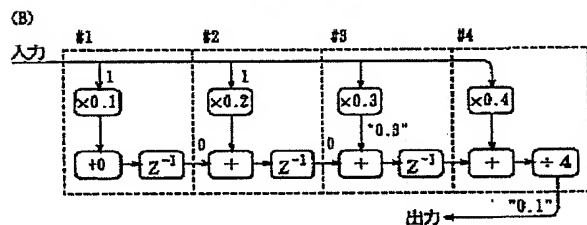
入力待ち

	B	C	A	X	D
1	1	0.1	0		0
2	1	0.2	0		0
3		0.3	0	0.3	0
4		0.4			2

×0.1, ×0.2, ×0.3, ×0.4

+0, Z<sup>-1</sup>, +, Z<sup>-1</sup>, +, Z<sup>-1</sup>, +, Z<sup>-1</sup>, +, ÷4

出力



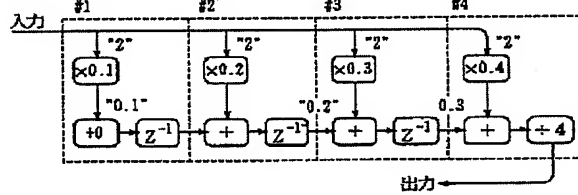
【図 46】

(A) 実行状態5

2を入力

	B	C	A	X	D
1	2	0.1	0	0.1	0
2	2	0.2			0
3	2	0.3	0.2		0
4	2	0.4	0.3		2

(B)



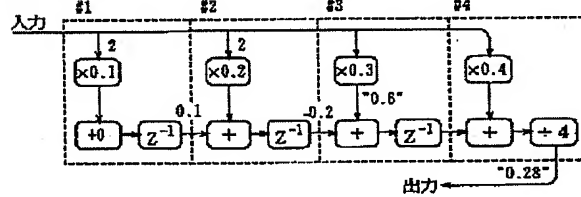
【図 48】

(A) 実行状態7

入力待ち

	B	C	A	X	D
1	2	0.1	0		0
2	2	0.2	0.1		0
3		0.3	0.2	0.6	0
4		0.4			2

(B)



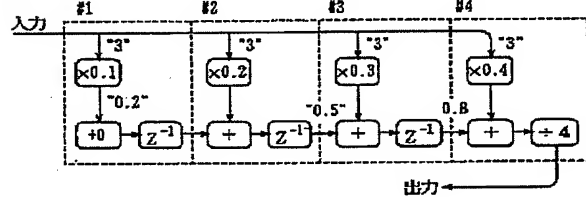
【図 50】

(A) 実行状態9

8を入力

	B	C	A	X	D
1	3	0.1	0	0.2	0
2	3	0.2			0
3	3	0.3	0.5		0
4	3	0.4	0.8		2

(B)



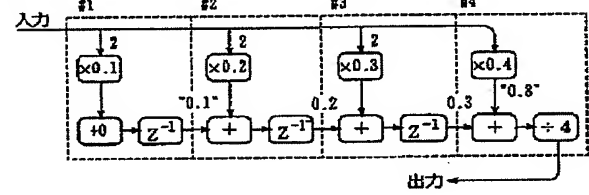
【図 47】

(A) 実行状態6

入力待ち

	B	C	A	X	D
1	2	0.1	0		0
2	2	0.2	0.1		0
3	2	0.3	0.2		0
4		0.4	0.3	0.8	2

(B)



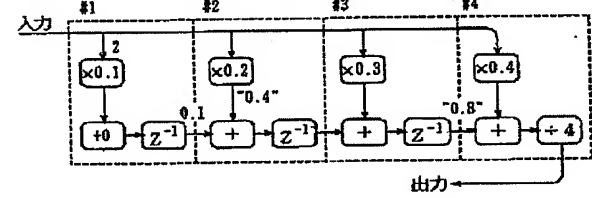
【図 49】

(A) 実行状態8

入力待ち

	B	C	A	X	D
1	2	0.1	0		0
2		0.2	0.1	0.4	0
3		0.3			0
4		0.4	0.8		2

(B)



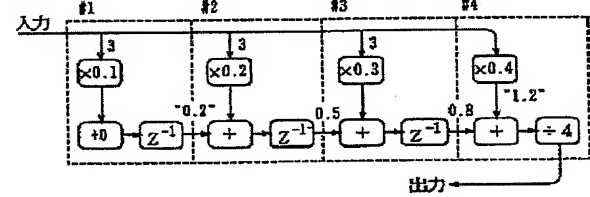
【図 51】

(A) 実行状態10

入力待ち

	B	C	A	X	D
1	3	0.1	0		0
2	3	0.2	0.2		0
3	3	0.3	0.5		0
4		0.4	0.8	1.2	2

(B)



【図52】

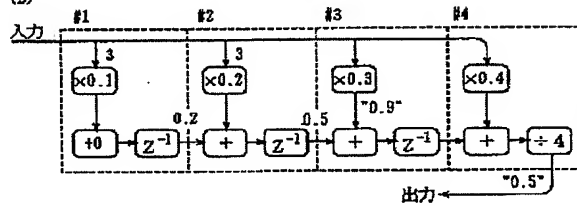
(A) 実行状態1.1

入力待ち

	B	C	A	X	D
1	3	0.1	0		0
2	3	0.2	0.2		0
3		0.3	0.5	0.9	0
4		0.4			2

※0.5出力値

(B)



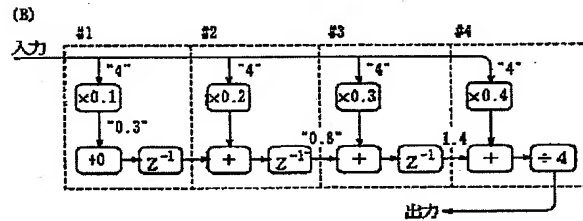
【図54】

(A) 実行状態1.3

4を入力

	B	C	A	X	D
1	4	0.1	0	0.3	0
2	4	0.2			0
3	4	0.3	0.8		0
4	4	0.4	1.4		2

(B)



【図56】

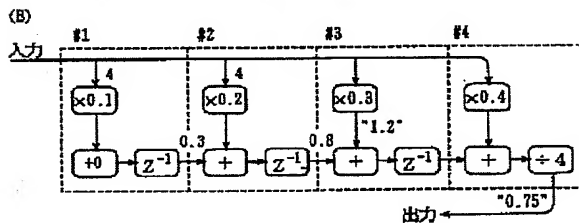
(A) 実行状態1.5

入力待ち

	B	C	A	X	D
1	4	0.1	0		0
2	4	0.2	0.3		0
3		0.3	0.8	1.2	0
4		0.4			2

※0.75出力値

(B)



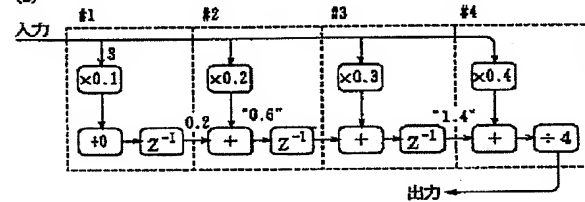
【図53】

(A) 実行状態1.2

入力待ち

	B	C	A	X	D
1	3	0.1	0		0
2		0.2	0.2	0.6	0
3		0.3			0
4		0.4	1.4		2

(B)



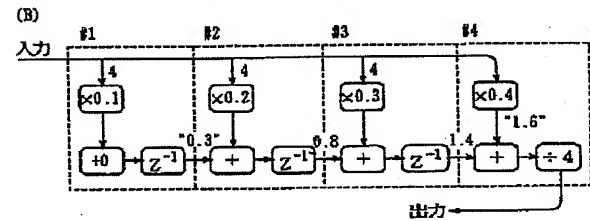
【図55】

(A) 実行状態1.4

入力待ち

	B	C	A	X	D
1	4	0.1	0		0
2	4	0.2	0.3		0
3	4	0.3	0.8		0
4		0.4	1.4	1.6	2

(B)



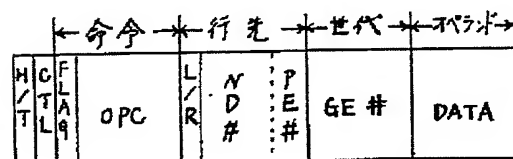
【図57】

サイクル2～16の各演算器の稼働状況

実行動作番号	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
データ入力															
× (乗算器)															
+shift (組合演算器)															
データ出力															

(○が稼働状態、空白がデータ待ち状態)

【図64】

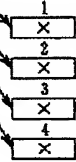


【図 59】

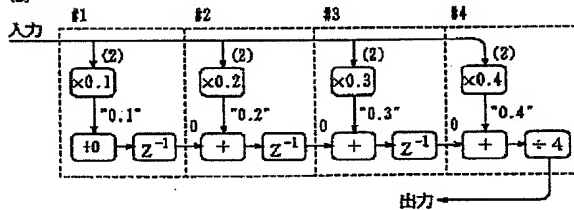
(A) 実行状態 1-b (1-aは演算器の動作なし)

BとCの値を×演算器に送り、その結果を(×の計算結果)に書き込む。

	B	C	A	X	D
1		0.1	0	0.1	0
2		0.2	0	0.2	0
3		0.3	0	0.3	0
4		0.4	0	0.4	2



(B)

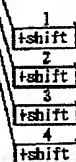


【図 60】

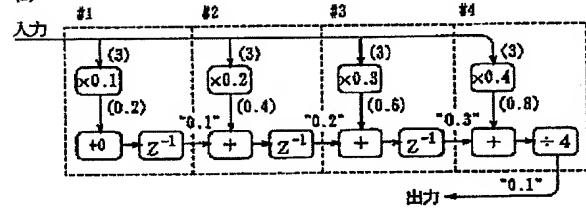
(A) 実行状態 2-a

(×の計算結果)とAの値を、レジスタの示す行き先に書き込む。この場合は、次のエントリ (1ならば2、2ならば3というように次の番号をもつレジスタ) のAレジスタ

	B	C	A	X	D
1		0.1	0	0.1	0
2		0.2	0.1	0.2	0
3		0.3	0.2	0.3	0
4		0.4	0.3	0.4	2

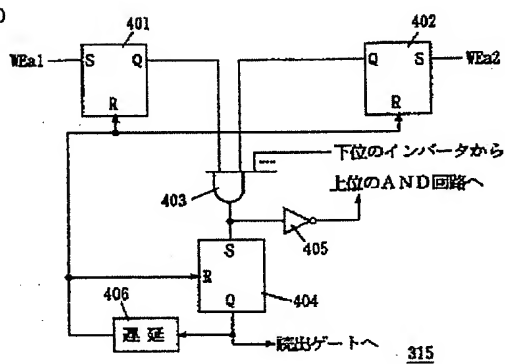


(B)



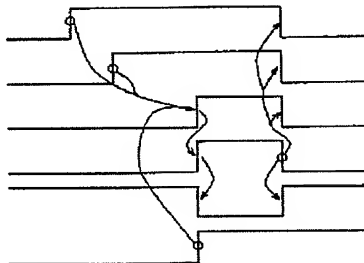
【図 62】

(A)

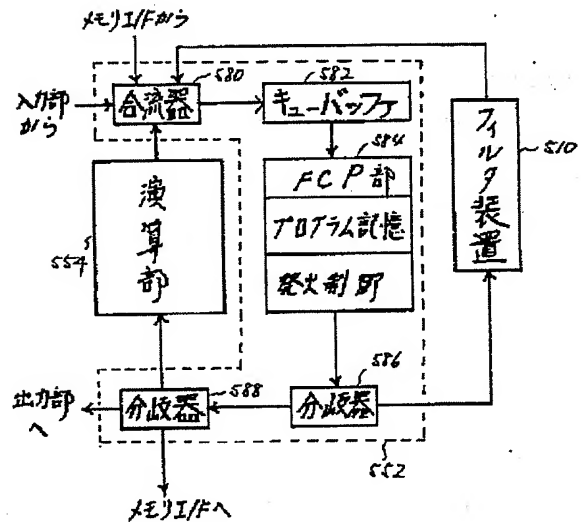


(B)

- (a) 401 出力  
(b) 402 出力  
(c) 403 出力  
(d) 404 出力  
(e) 405 出力  
(f) 下位インバータ

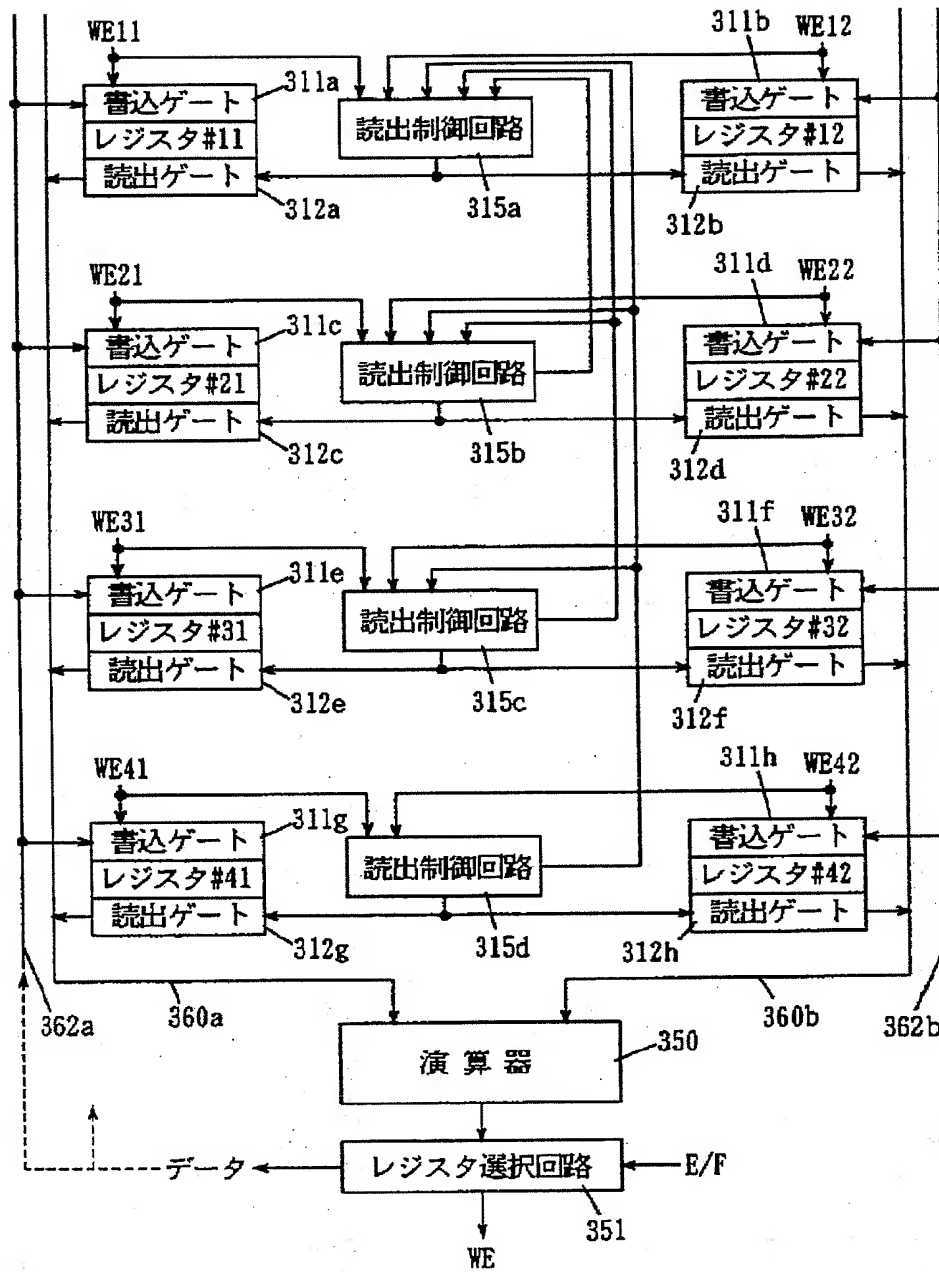


【図 63】

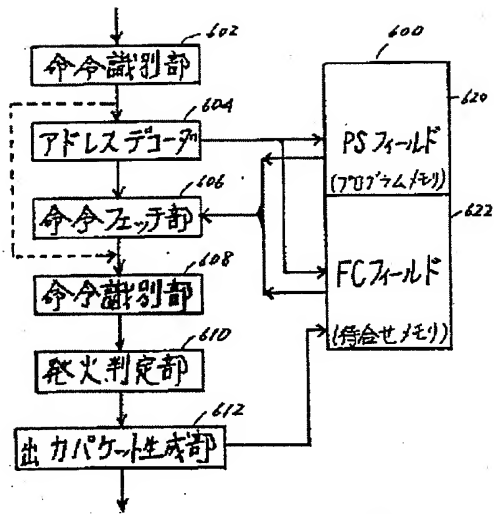




【図61】



【図65】



584

【図68】

